

《【曾贤志】从零基础开始用 Python 处理 Excel 数据》对应视频链接如下：

<http://edu.51cto.com/sd/e9508>

目录

第 1 章 python 基础.....	5
1.1 什么是 python?	5
1.2 为什么要学习用 Python 处理 Excel 表格?	5
1.3 手把手教你安装 python 程序.....	6
1.3.1 下载 python.....	6
1.3.2 安装 python.....	6
1.3.3 验证是否安装成功.....	9
1.4 安装 Python 集成开发工具 PyCharm.....	10
1.4.1 下载.....	11
1.4.2 安装.....	11
1.5 Python 的输入与输出.....	17
1.5.1 Print 输出.....	17
1.5.2 Input 输入.....	18
1.6 Python 的代码注释.....	18
1.6.1 单行注释.....	18
1.6.2 多行注释.....	19
1.7 学 Python，不愁没对象!	19
1.7.1 对象的身份.....	20
1.7.2 对象的类型.....	20
1.7.3 对象的值.....	20
1.7.4 对象的属性和方法.....	20
1.7.5 对象与变量.....	20
1.8 Python 中的数字与字符串.....	21
1.8.1 数字.....	21
1.8.2 字符.....	21
1.9 算术运算符.....	21
1.9.1 加 (+).....	21
1.9.2 减 (-).....	22
1.9.3 乘 (*).....	22
1.9.4 除 (/).....	22
1.9.5 取模 (%).....	22
1.9.6 幂 (**).....	22
1.9.7 取整数 (//).....	22
1.10 比较运算符.....	22
1.10.1 等于 (==).....	22
1.10.2 不等于 (!=).....	23

1.10.3 大于 (>)	23
1.10.4 小于 (<)	23
1.10.5 大于等于 (>=)	23
1.10.6 小于等于 (<=)	23
1.11 赋值运算符	23
1.11.1 赋值运算	23
1.11.2 累积式赋值运算	24
1.12 逻辑运算符	24
1.12.1 and (与)	24
1.12.2 or (或)	25
1.12.3 not (非)	25
1.13 成员运算符	25
1.13.1 成员运算符 (in)	25
1.14 格式化字符串	25
1.15 完美看清代码运行过程	26
第 2 章 用 Python 对 Excel 读写数据	28
2.1 什么是模块、包、库	28
2.2 安装 Excel 读取库 xlrd	28
2.3 xlrd 模块导入	30
2.4 读取 Excel 工作簿、工作表信息	30
2.5 读取 Excel 行、列、单元格信息	30
2.6 安装 Excel 写入库 xlwt	31
2.7 创建工作簿、工作表和写入单元格	32
2.8 安装 Excel 修改库	32
2.9 修改工作簿、工作表、单元格	34
第 3 章 循环语句与分支语句	35
3.1 for...in 循环语句基础	35
3.1.1 循环字符串	35
3.1.2 循环指定范围序列数	36
3.2 for...in 循环语句应用 (批量创建工作簿)	36
3.3 for...in 嵌套循环语句	36
3.4 for...in 嵌套循环语句应用 (制作九九乘法表)	37
3.5 while 循环语句	37
3.6 while 循环语句应用 (读取工作簿信息到新表)	38
3.7 while 嵌套循环语句	38
3.8 while 嵌套循环语句应用 (批量创建工作簿、工作表)	39
3.9 if 条件语句	39
3.10 if...else 条件语句	40
3.11 多条件 if 语句 1 (将筛选结果写入新工作簿)	40
3.12 多条件 if 语句 2 (根据总分判断等级)	41
3.13 break 语句 (跳出整个循环)	42
3.14 continue 语句 (跳出当次循环)	43
3.15 综合应用	43
第 4 章 字符串处理技术	45

4.1 字符串切片	45
4.1.1 提取单个字符	45
4.1.2 提取多个字符	45
4.2 字符串切片应用（整理工作表数据）	46
4.3 字符串长度（个数）统计	47
4.3.1 len	47
4.3.2 count	47
4.4 字符串的查找	48
4.4.1 index	48
4.4.2 find	48
4.5 字符串的替换	49
4.6 字符串的拆分与合并	50
4.6.1 split	50
4.6.2 join	50
第 5 章 列表处理技术	52
5.1 列表基础	52
5.2 列表切片	52
5.3 列表的增加、删除、修改	52
5.3.1 列表元素的修改	53
5.3.2 列表元素的增加	53
5.3.3 列表元素的删除	53
5.4 实例应用（汇总每个人的总成绩）	53
5.5 列表操作符	54
5.5.1 操作符基础	55
5.5.2 实例应用：判断指定姓名在指定日期是否值班	55
5.6 列表推导式	55
5.7 实例应用（筛选各工作表中符合条件的值）	56
5.8 列表转换	56
5.9 实例应用（统计出大于等于 2 万的记录到新表）	57
5.10 列表常见统计方式 1	58
5.11 列表常见统计方式 2	59
第 6 章 元组处理技术	60
6.1 元组的创建	60
6.2 元组的基本操作	60
6.3 元组常用统计方法	61
第 7 章 字典处理技术	62
7.1 字典的基础	62
7.1.1 创建字典	62
7.1.2 字典键的特性	62
7.1.3 获取字典里的值	63
7.1.4 小实例	63
7.2 实例应用（提取产品最后的记录）	63
7.3 字典的转换	63
7.3.1 dict 转换法	64

7.3.2 dict.fromkeys 转换法	64
7.4 实例应用（多列求唯一值）	64
7.5 字典的删除	65
7.5.1 Clear 方法清空字典	65
7.5.2 pop 方法删除字典	65
7.5.3 del 语句删除字典	66
7.6 实例应用（查询未发货订单）	66
7.7 字典的修改	67
7.7.1 字典的值修改	67
7.7.2 字典的键名修改	67
7.7.3 字典的连接	68
7.8 实例应用（统计各种蔬菜的总金额）	68
7.9 字典的查询	68
7.9.1 判断是否存在	68
7.9.2 单值查询	69
7.9.3 多值查询	69
7.10 实例应用（统计各日期的销售数据）	69
7.11 字典的循环	70
7.11.1 普通循环	70
7.11.2 字典在列表中的推导	71
7.11.3 字典推导式	71
7.12 实例应用（统计各省各公司总业绩）	71
第 8 章 集合处理技术	73
8.1 集合的创建	73
8.1.1 可变集合(set)	73
8.1.2 不可变集合(frozenset)	73
8.2 实例应用（判断指定项目是否存在）	73
8.3 集合的添加与删除	74
8.3.1 集合的添加	74
8.3.1 集合的删除	74
8.4 实例应用（多行多列求唯一值）	75
8.5 集合的循环与推导	75
8.5.1 集合的循环	75
8.5.2 集合的推导	76
8.6 实例应用（统计每月每个战队的人数）	76
8.7 集合的运算	76
8.8 实例应用（求每个人不达标的月份）	77
第 9 章 Python 函数技术	79
9.1 自定义函数的基本结构	79
9.1.1 函数的定义	79
9.1.2 实例应用（定义平均函数及应用）	79
9.2 位置参数写法及应用	80
9.2.1 位置参数（必需参数）	80
9.2.2 实例应用（等级判断函数及应用）	80

9.3 默认参数写法及应用	81
9.3.1 默认参数（可选参数）	81
9.3.2 实例应用（定位指定字符串的位置）	81
9.4 关键字参数写法及应用	82
9.4.1 关键字参数	82
9.4.2 实例应用（汇总字符串中指定位置的数字）	83
9.5 不定长参数写法及应用	83
9.5.1 不定长参数	83
9.5.2 实例应用（增强替换函数的功能）	84
9.6 匿名函数写法及应用	84
9.6.1 匿名函数	85
9.6.2 实例应用：根据身份证号判断性别	85
9.7 递归函数写法及应用	86
第 10 章 高阶函数应用	87
10.1 map 转换函数	87
10.1.1 map 函数的应用方法	87
10.1.2 实例应用（每个月大于等于 100 的记录数）	87
10.2 filter 筛选函数	88
10.3 Sort 与 Sorted 排序函数	90
10.3.1 sort 方法	90
10.3.2 sorted 函数	90
10.3.3 实例应用（自定义排序）	91

第1章 python 基础

1.1 什么是 python？

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347192>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。被称为胶水语言，Python 应用场景广泛，比如 WEB 开发、网络编程、爬虫、云计算、人工智能、自动化运维、数据分析、游戏开发等……好像有点无所不能。

1.2 为什么要学习用 Python 处理 Excel 表格？

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347192>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Excel 是一款非常平民化，但功能又强大的电子表格软件。但再强大的软件也有短板，

比如要批量的、重复性的做一些事，或者做数据分析，就必须用 Excel 中的 VBA 来做二次开发，VBA 虽然强大，但 VBA 相比 Python，Python 的应用范围更加广泛、代码更加整洁、更简单一些，唯一麻烦一点的是需要独立安装。

学习 Python 处理 Excel 表格后，也可以利用 Python 做更多的事，你也可以理解成我们是以 Excel 为切入点来学习 Python 这门编程语言。

1.3 手把手教你安装 python 程序

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347190>

全套视频链接：<http://edu.51cto.com/sd/e9508>

在学习 Python 之前，需要安装 Python 解释器，由于我们使用的是 Windows 系统，所以下载的 Python 解释器必须是支持 Windows 系统的。

1.3.1 下载 python

要使用 python 进行编程，首先要下载这个程序，首先当然是到官方网站 (<https://www.python.org>) 下载，输入网站之后，效果如图 1.3-1 所示，然单击【Downloads】菜单→【Windows】命令→【Download Windows x86-64 executable installer】，我们选择的是 Python 7.7.1 版本 32 位的 Python，使用者可根据自己的实际情况选择不同的版本。

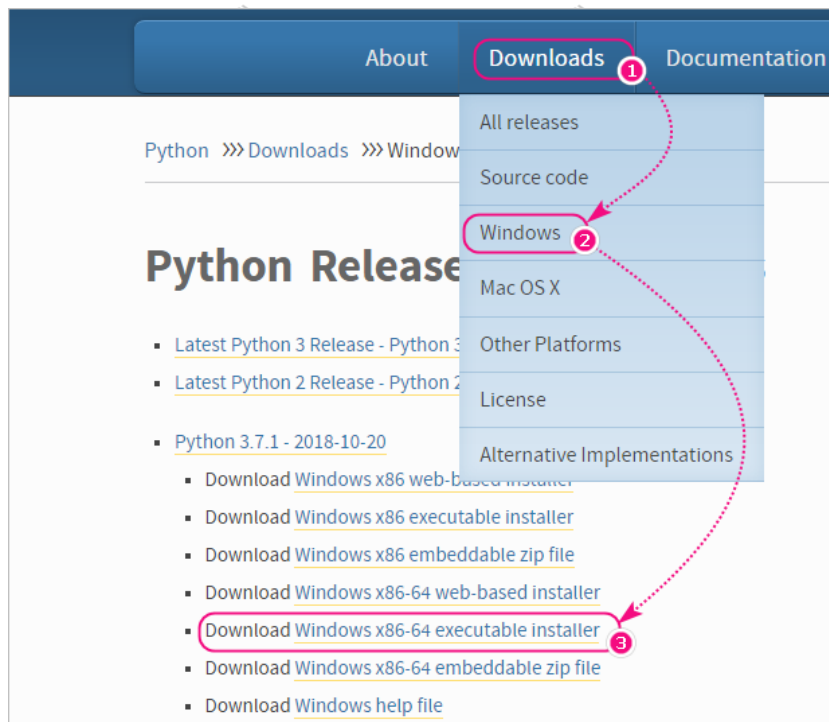


图 1.3-1

1.3.2 安装 python

下载好 python 软件后，下面开始安装。

第 1 步：鼠标指向 python 软件包，然后单击右键，在弹出的快捷菜单中单击【以管理员身份运行】，如图 1.3-2 所示。



图 1.3-2

第 2 步：如图 1.3-3 所示，在弹出的对话框中首先勾选【Install launcher for all user(recommended)】和【Add Python 3.7 to PATH】，然后再单击【Customize installation】，执行自定义安装。

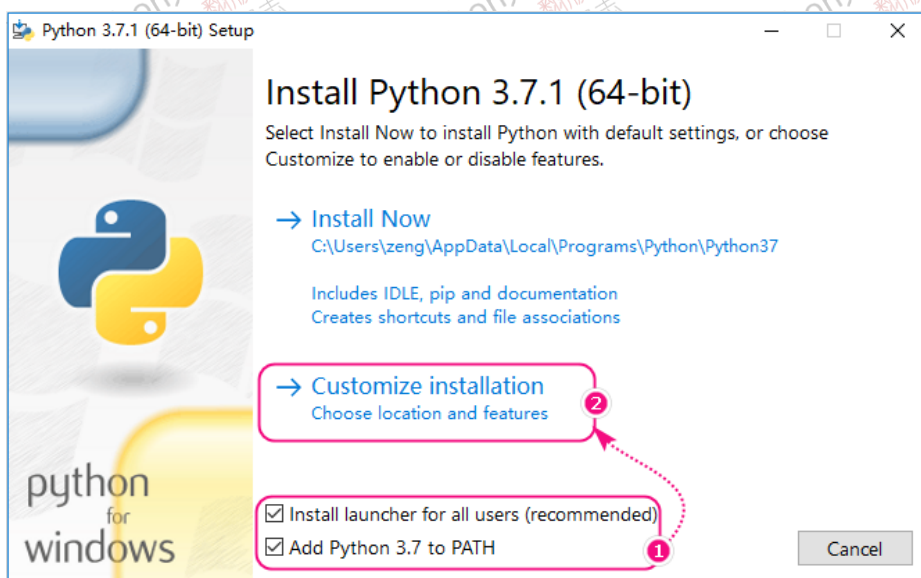


图 1.3-3

第 3 步：如图 1.3-4 所示，在弹出的对话框中勾选所有的复选框，然后再单击【Next】按钮，继续执行后续安装。

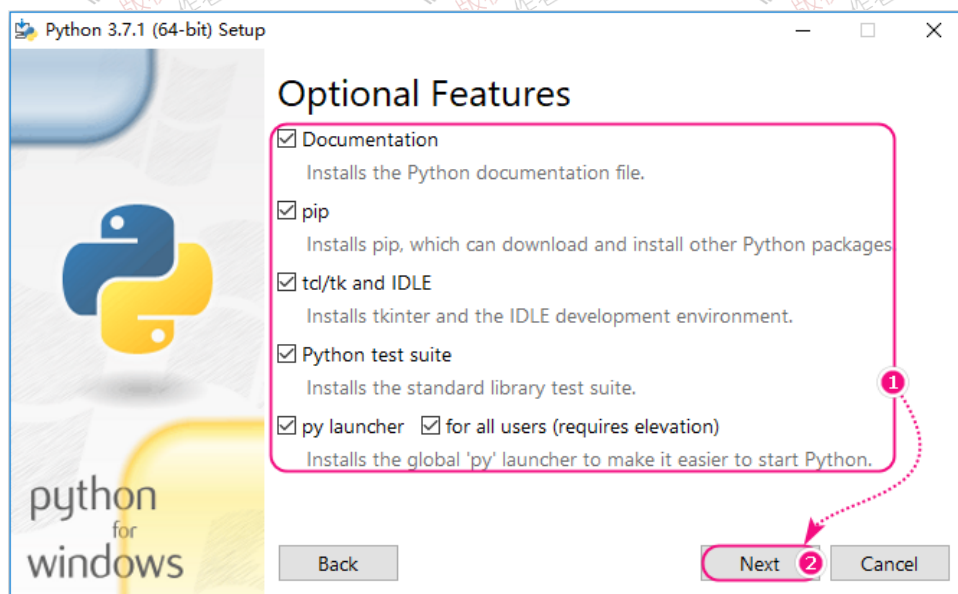


图 1.3-4

第 4 步：如图 1.3-5 所示，在弹出的对话框中勾选所有的复选框，在单击【Browse】按钮，选择程序安装的位置，本示例选择的是“D:\python”文件夹，最后单击【Install】按钮开始安装。

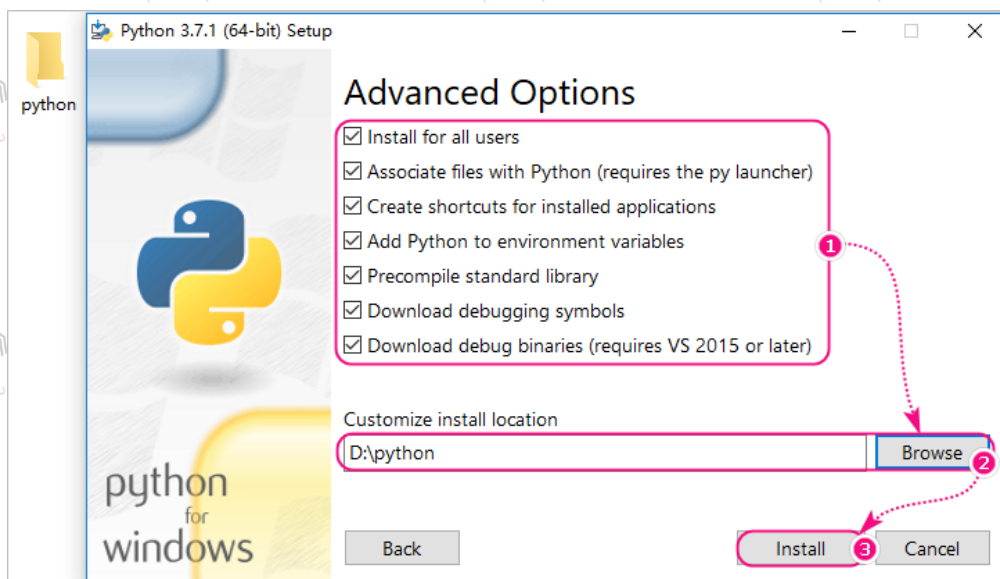


图 1.3-5

第 5 步：如图 1.3-6 所示，正在安装 Python 程序，只需要等待即可。

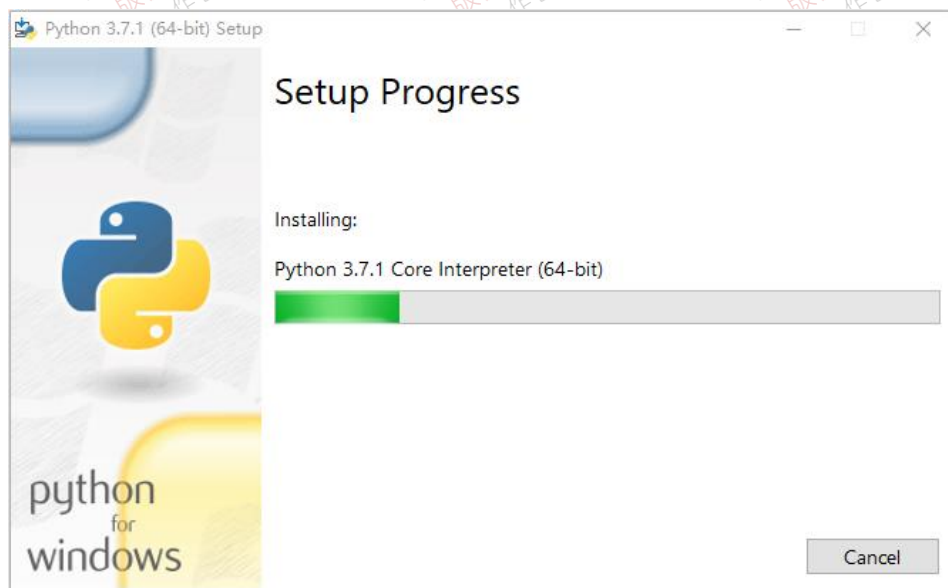


图 1.3-6

第 6 步: 如图 1.3-7 所示, 提示“Setup was successful”, 表示安装成功, 最后单击【Close】按钮即可。

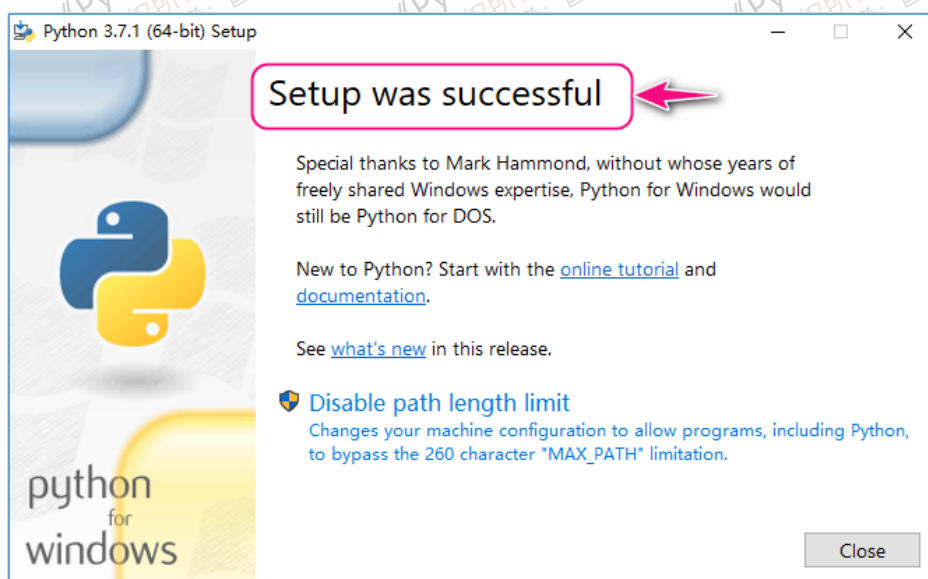


图 1.3-7

1.3.3 验证是否安装成功

在安装完成 python 程序之后, 可以做个测试来判断是否安装成功, 操作方法如下。

第 1 步: 如图 1.3-8 所示, 鼠标指向 Windows 系统左下角的开始按钮, 然后单击鼠标右键, 在弹出来的快捷菜单中单击选择【运行】。还有一种快捷方式, 就是按组合键: **Win** (Windows 键) + **R**。

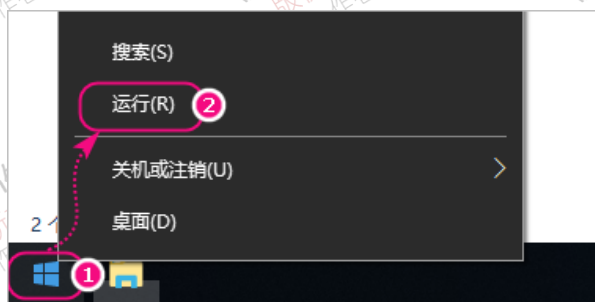


图 1.3-8

第 2 步：如图 1.3-9 所示，在弹出的【运行】对话框中，在【打开】文本框中输入命令：“cmd”，然后再单击【确定】按钮。

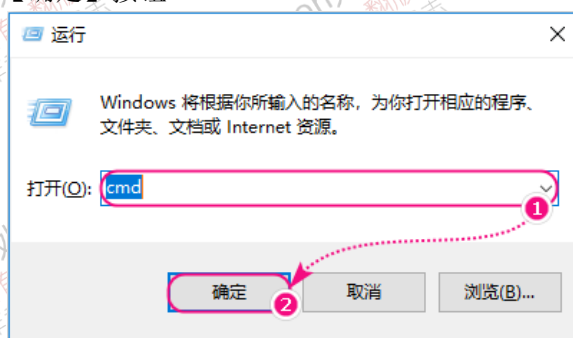


图 1.3-9

第 3 步：如图 1.3-10 所示，在命令提示符对话框中输入“python”，再按回车键。



图 1.3-10

第 4 步：如图 1.3-11 所示，显示了与 Python 相关的版本信息，进入 Python 的交互模式，表明安装成功。

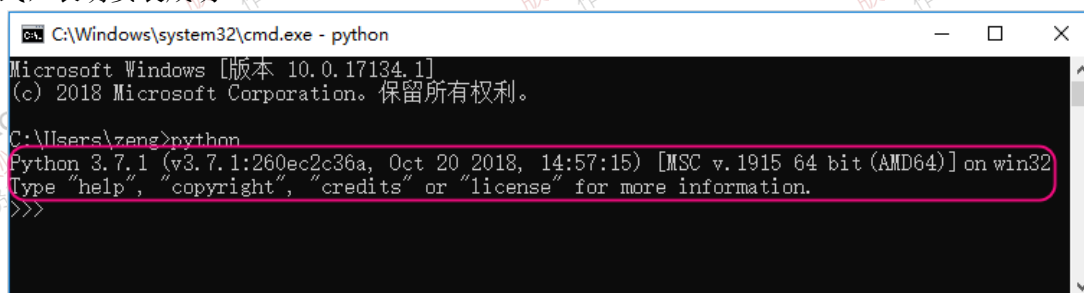


图 1.3-11

1.4 安装 Python 集成开发工具 PyCharm

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347189>

全套视频链接：<http://edu.51cto.com/sd/e9508>

PyCharm 是一种 Python IDE，带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具，比如调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。

1.4.1 下载

第 1 步：输入网址 <https://www.jetbrains.com/pycharm/>，然后弹出如图 1.4-1 所示界面，再单击【DOWNLOAD NOW】按钮。

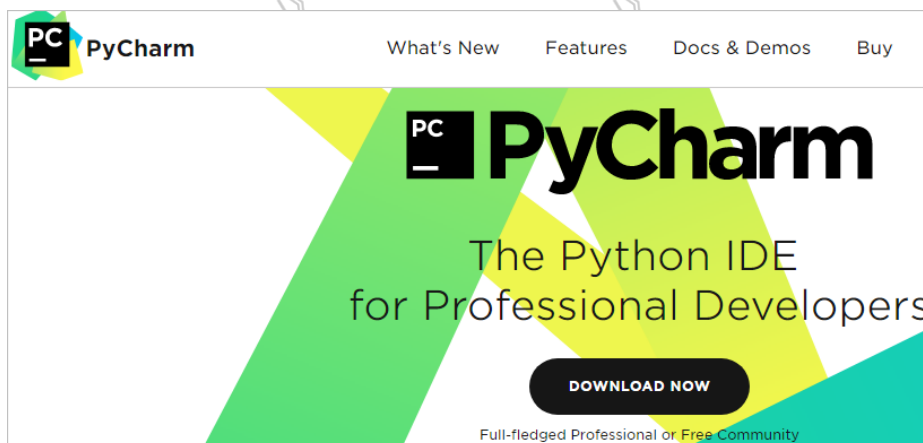


图 1.4-1

第 2 步：在弹出的页面中，单击【Windows】按钮，有 Professional（专业版|收费）和 Community（社区版|免费），所以单击 Community 下的【DOWNLOAD】按钮进行下载即可。

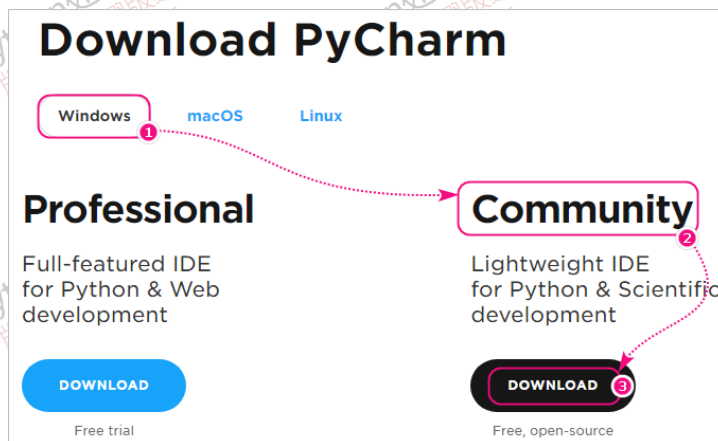


图 1.4-2

1.4.2 安装

下载好 Python 编辑器 Pycharm 软件之后，开始安装，请按照安装要求逐步进行安装设置。

第 1 步：如图 1.4-3 所示，鼠标指向【pycharm-community-2018.3】，然后单击右键，在弹出的快捷菜单中选择【以管理员身份运行】即可。



图 1.4-3

第 2 步：如图 1.4-4 所示，单击【Next】按钮，继续进行安装。

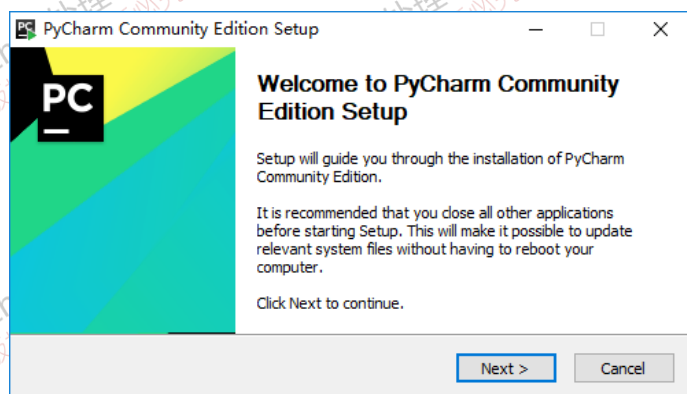


图 1.4-4

第 3 步：如图 1.4-5 所示，单击【Browse...】按钮，选择 Pycharm 软件安装的位置，然后再单击【Next】按钮，继续进行安装设置。

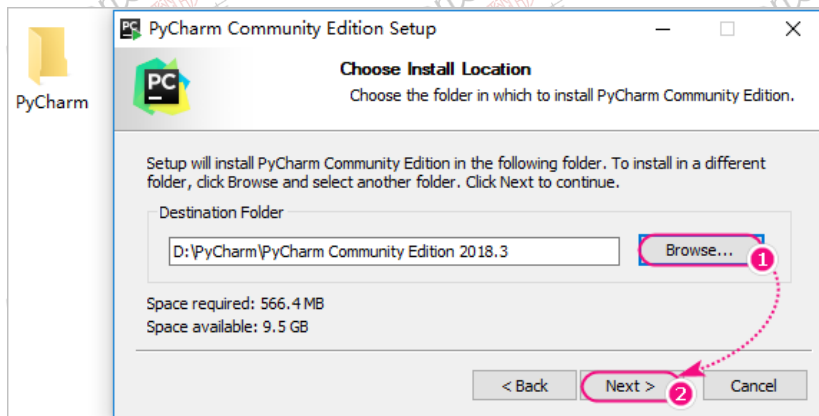


图 1.4-5

第 4 步：如图 1.4-6 所示，如果操作系统是 64 位则勾选【64-bit launcher】，如果是 32 位则勾选【32-bit launcher】，再勾选【Add launchers dir to the PATH】、【Add “Open Folder as Project”】、【.py】几个复选框，最后单击【Next】按钮，继续进行安装设置。

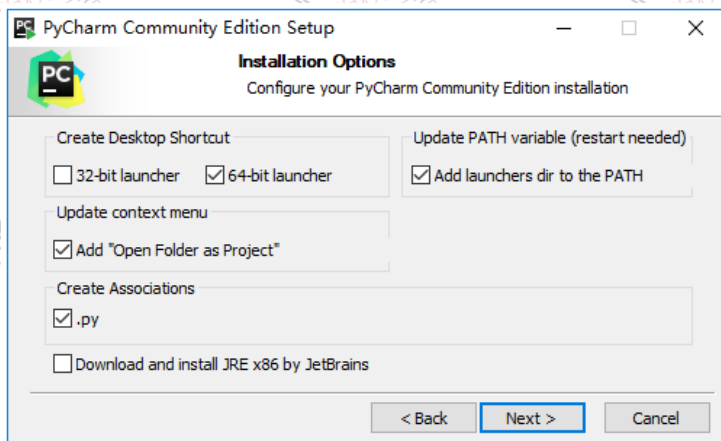


图 1.4-6

第 5 步：如图 1.4-7 所示，做好 Pycharm 前面的相关设置之后，现在开始正式安装程序，所以单击【Install】按钮，开始正式安装。

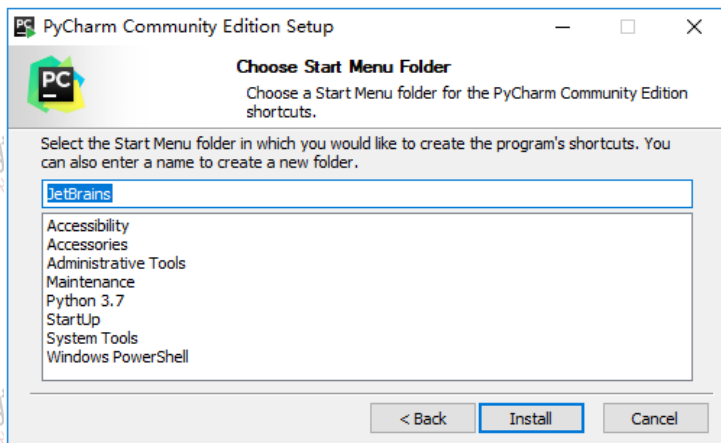


图 1.4-7

第 6 步：如图 1.4-8 所示，等待安装即可。

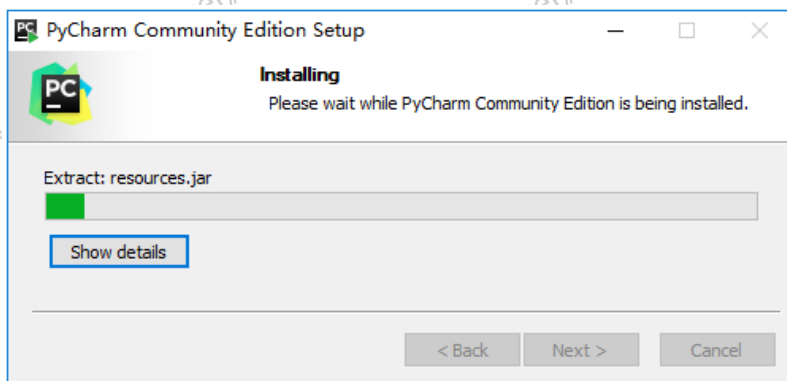


图 1.4-8

第 7 步：如图 1.4-9 所示，安装完成之后，再选择【Reboot Now】，表示要重新启动系统，单击【Finish】按钮即可。

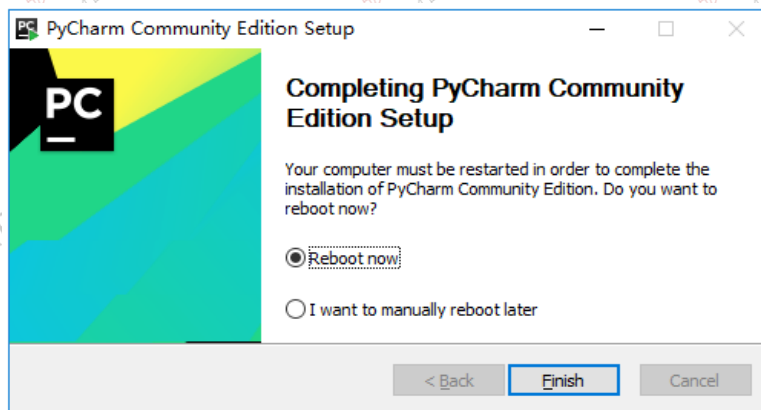


图 1.4-9

第 8 步: 如图 1.4-10 所示, 重启操作系统之后, 双击打开【JetBrains PyCharm Community Edition 2018.3 x64】程序, 然后选择【Do not import setting】, 再单击【OK】按钮, 继续做设置。

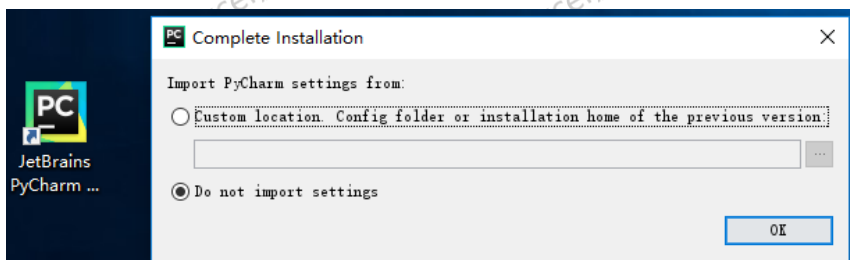


图 1.4-10

第 9 步: 如图 1.4-11 所示, 勾选【I confirm that I have read and accept the terms of this User Agreement】复选框, 再单击【Continue】按钮。

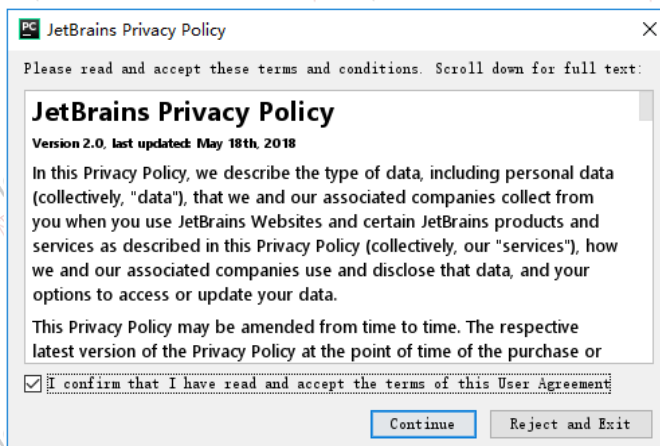


图 1.4-11

第 10 步: 如图 1.4-12 所示, 单击【Don't send】按钮, 继续进行设置。

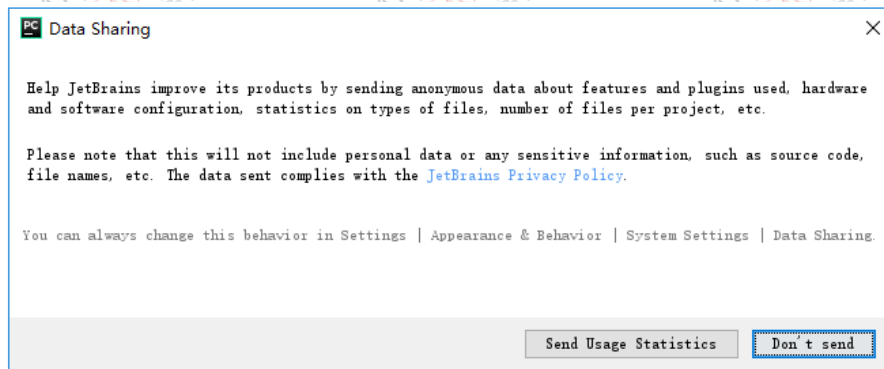


图 1.4-12

第 11 步: 如图 1.4-13 所示, 选择一种风格, 在这里选择的是【Darcula】, 再单击【Skip Remaining and Set Defaults】按钮, 默认设置即可。

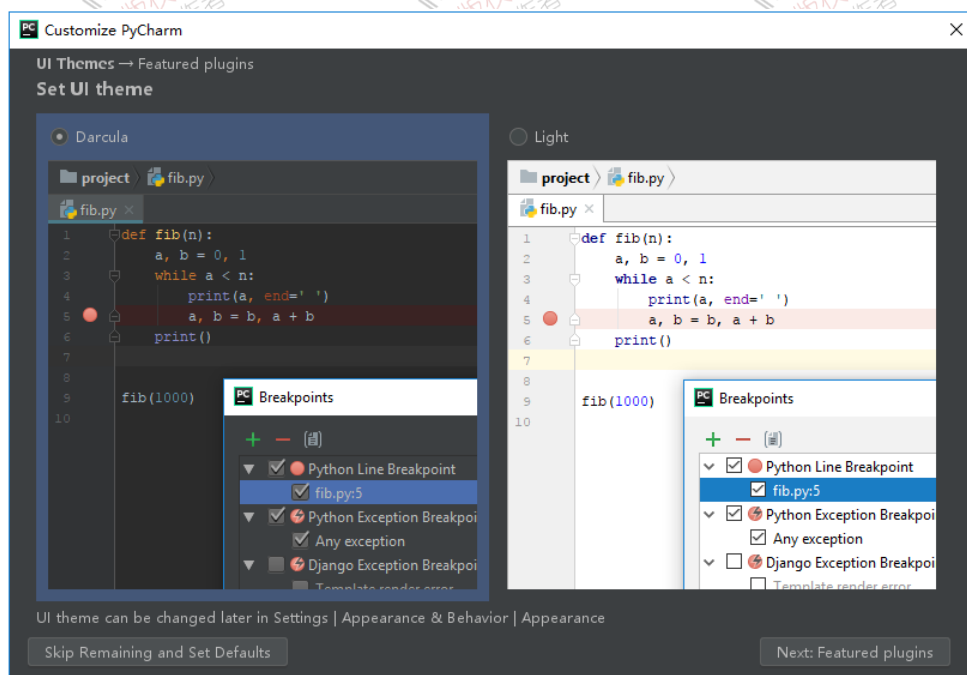


图 1.4-13

第 12 步: 如图 1.4-14 所示, 等待 PyCharm 设置即可。



图 1.4-14

第 13 步：如图 1.4-15 所示，现在开始创建工程，单击【Create New Project】选项即可。

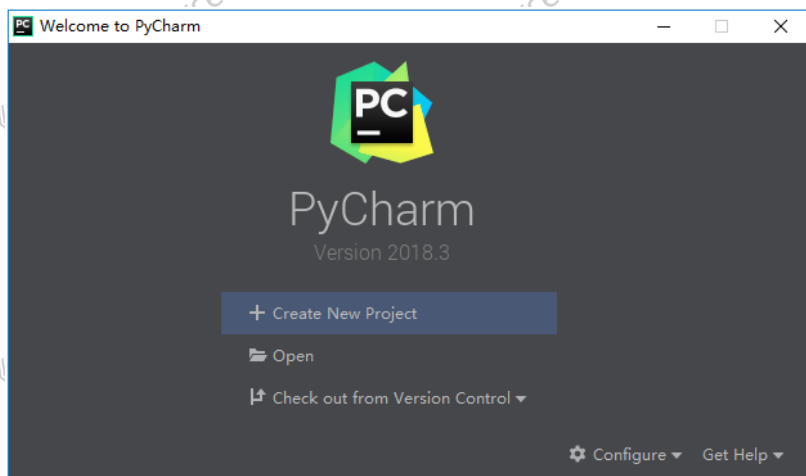


图 1.4-15

第 14 步：如图 1.4-16 所示，选择新建工程要放置的目录，这里选择的是 D 盘的 Project 文件夹，然后单击【Create】按钮即可。

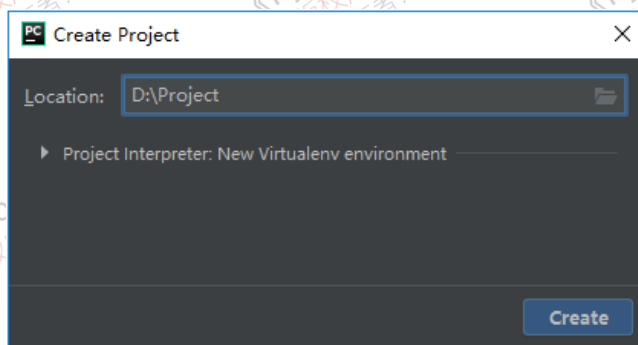


图 1.4-16

第 15 步：如图 1.4-17 所示，对话框中显示的是 PyCharm 欢迎界面，关闭即可，所以单击【Close】按钮即可。

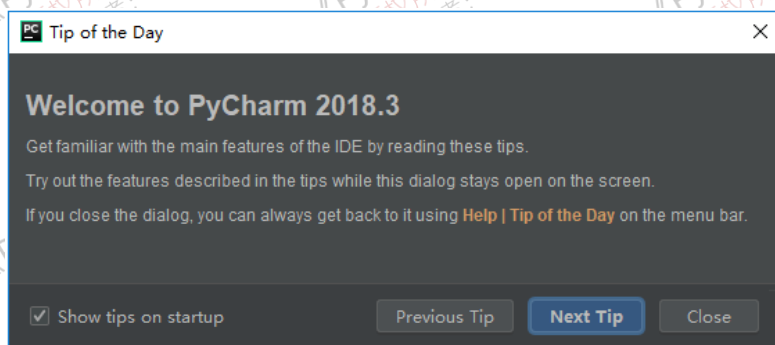


图 1.4-17

第 16 步：如图 1.4-18 所示，鼠标指向 Project 文件夹，然后单击右键，在弹出的快捷菜单中单击【New】列表→【Python File】命令即可。

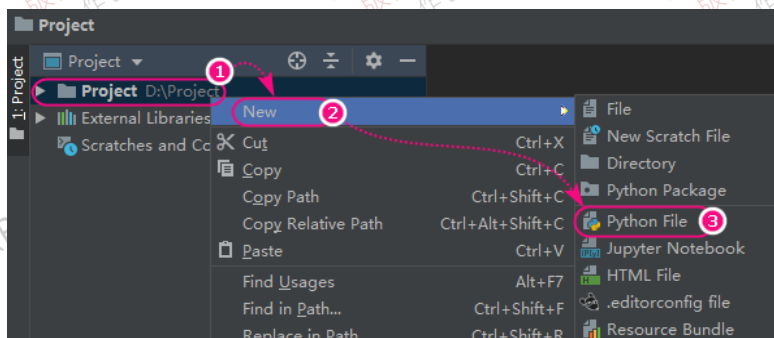


图 1.4-18

第 17 步：如图 1.4-19 所示，在弹出的对话框中，将 python 的文件名称命名为“test”，然后单击【OK】按钮即可。

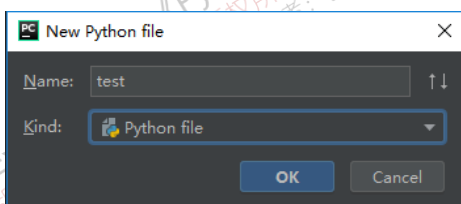


图 1.4-19

第 18 步：如图 1.4-20 所示，在 test.py 文件中输入代码“print('Hello,Python!')”，然后单击  按钮运行，在屏幕上打印出字符串“Hello,Python!”。说明安装一切安装成功！至此我们可以利用 PyCharm 正式学习 Python 了。

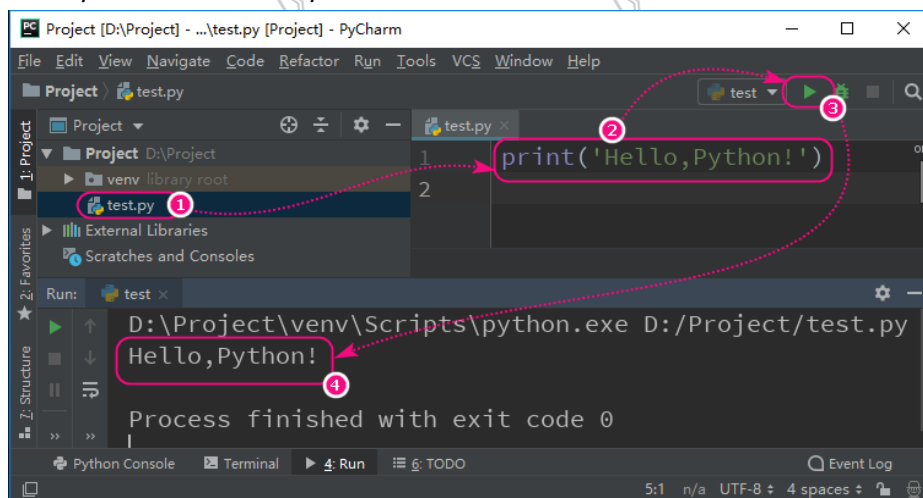


图 1.4-20

1.5 Python 的输入与输出

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347188>


全套视频链接：<http://edu.51cto.com/sd/e9508>

1.5.1 Print 输出

当安装好 python 软件和 Pycharm 编辑器之后，我们就迫不及待想试试身手，那就从 print

开始吧！Print 是一个打印函数，将想要显示的数据打印在屏幕上。在编辑器中输入：

```
1-> print('嗨！python 我来了！')
```

点击运算  按钮的效果如图 1.5-1 所示：

```
嗨！python我来了！
```

```
Process finished with exit code 0
```

图 1.5-1

1.5.2 Input 输入

终于学会了第一条代码，使用 print 来完成在 python 中数据的输出。那如果要向 python 中输入语句呢？请看如下代码：

```
1-> name=input('请输入你的姓名：')  
2-> print(name)
```

上面的代码有两句，如图 1.5-2 所示，运行第 1 句时，屏幕会提示'请输入你的姓名：' 提示语，并等待用户输入数据。当用户输入数据后，将得到的数据贴上标签 name。运行第 2 句时，则是将 name 标签对应的数据使用 print 打印出来。运行结束。

```
请输入你的姓名：曾贤志
```

```
曾贤志
```

```
Process finished with exit code 0
```

图 1.5-2

1.6 Python 的代码注释

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347187>

全套视频链接：<http://edu.51cto.com/sd/e9508>

所谓代码注释，就是为写好的代码添加说明。对代码做注释有以下几点好处：

1. 方便自己查看写过的代码
2. 互联网人才流动大，为了项目能更好维护，需要适当写注释，其他程序员能更快的读懂代码作用。
3. 在编写代码时，还可以做代码调试。

1.6.1 单行注释

单行注释，只作用到一行。可以在单独的行写注释，在写注释内容之前，首先输入“#”（井号），如下代码所示。


```
1-> #将数据打印到屏幕上
2-> print('嗨！python 我来了！')
3-> #在此输入姓名
4-> name=input('请输入你的姓名：')
5-> #将打印出输入的数据
6-> print(name)
```

单行注释也可以写在代码的后面。同样，也要先输入“#”号，然后再输入内容，所下代码所示。

```
1-> print('嗨！python 我来了！') #将数据打印到屏幕上
2-> name=input('请输入你的姓名：') #在此输入姓名
3-> print(name) #将打印出输入的数据
```

1.6.2 多行注释

如果有大段的注释文字要写，则可以使用多行注释的方法。多行注释的内容只要包含在一对单引号中即可，6 个单引号为一对，如下所示代码：

```
1-> '''
2-> print 函数表示将数据打印出来
3-> input 函数表示要接收输入的数据
4-> 两个函数可以结合使用。
5-> '''
6-> name=input('请输入你的姓名：')
7-> print(name)
```

除了使用单引号，也可以使用双引号来注释，如下所示代码：

```
1-> """
2-> print 函数表示将数据打印出来
3-> input 函数表示要接收输入的数据
4-> 两个函数可以结合使用。
5-> """
6-> name=input('请输入你的姓名：')
7-> print(name)
```

到底是使用单引号还是双引号来做注释，没有强制规定，完全根据使用者的习惯或者好恶来定，想咋的就咋的。

1.7 学 Python，不愁没对象！

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347186>

全套视频链接：<http://edu.51cto.com/sd/e9508>

真实世界是由千千万万的对象组成的。人畜虾蟹、牛鬼蛇神都是这世界上一种对象，所以万物皆对象。在 python 的编程世界里，所有的一切也可以看做对象。比如数字、字符，以及后面将会学到的列表、元组、字典、函数……，都是对象，你可以使用这些对象，也可以自己创造对象。

1.7.1 对象的身份

人是现实世界中的对象，可以靠身份证号来识别，可以定位具体住在哪里，python 中的对象是有身份的。可以通过 id 函数来识别对象在内存中的地址。比如输入代码“`print(id('曾贤志'))`”输出结果为“2673213256496”，这串数据就可以看做在内存中的位置。

1.7.2 对象的类型

虽然万物皆对象，但对象也有类别之分。比如猪狗牛马、花草树木等都是对象，但它们却是不同类型，猪狗牛马是动物类型，花草树木是植物类型。

比如在 python 中“99、888、abc”都是对象，“9”和“888”是数字类型，而“abc”是字符串类型。不同的类型有着不同的属性和方法，遵循不同的规则，有着不同的操作方法。

要查看对象的类型可以使用 type 函数，比如输入代码“`print(type(99))`”，返回结果为“<class 'int'>”，表示 99 是 int 类型，也就是整数的意思。再比如输入代码“`print(type('abc'))`”，返回结果为“<class 'str'>”，表示“abc”是 str 类型，也就是字符串的意思。其它对象的类型就不在这里一一例举，后面将会慢慢学习到。

1.7.3 对象的值

对象除了有身份证、类型，还有值，就像人的名字就可以看做是值。在 Python 中有的值是可以改变的，有的值是不可改变的，比如数字、字符、元组都是不可变的对象。

1.7.4 对象的属性和方法

在 python 中，一个对象的特征也称为属性，它所具有的行为也可以称作方法（函数）。我们可以通过属性和方法来操作对象。

1.7.5 对象与变量

在编程过程中，很多时候需要对对象使用变量更方便。所谓变量，相当于给创建的对象贴个标签，这样更方便识别。

比如“`a=1`”，表示的意思是给对象“1”贴个标签，当引用变量 a 进行代码编写时，就相当于在使用对象“1”。

再说说变量命名的规则，变量名可以是数字、字母、下划线几种元素的任意组合。但需

要注意的是，变量名不能以数字开始，也不能使用 python 中的关键字，变量名尽量取得有意义。

1.8 Python 中的数字与字符串

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347185>

全套视频链接：<http://edu.51cto.com/sd/e9508>

前面我们说过，python 中万物皆对象。数字与字符串也只是其中的两种对象，我们先讲解这两种比较好学习的对象，后面将陆续讲解更多对象。

1.8.1 数字

Python 中的数字有三种类型：整数、浮点数（小数）、复数 3 种。有时需要对数据进行转换，可以使用对象的函数，转为整数使用 int 函数，转为小数使用 float 函数，转为复数使用 complex 函数。

数字也可以与算术运算符结合进行算术运算。

1.8.2 字符

所谓字符串，也就是一串字符，一串字符，也就是一个及以上的字符集合。Python 中的字符串必须被一对单引号（'）或者双引号（"）包围。如果将其它数据转为字符串类型，可以使用 str 函数。

在 python 中，还有一些常用的特殊字符。比如换行符（\n）、制表符（\t）、回车符（\r）等等。如果遇到特殊字符，需要将其转为普通字符时在前面加上\即可。还有另一种表示方法，则是在字符串的左外侧加上字母 r（大小写均可）。比如代码：“print('我是谁！\n 我在哪儿！')”其中有\n 表示要换行。如果表示成：“print(r'我是谁！\n 我在哪儿！')”则\n 没有换行效果了，只是普通字符。

1.9 算术运算符

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347184>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Python 中的算术运算符有加（+）、减（-）、乘（*）、除（/）、取模（%）、幂（**）、取整数（//）。

1.9.1 加（+）

加运算符也可以执行数字的算术运算符，比如，“print(100+199)”，将返回 299。也可以进行字符串的连接运算，比如，“print('曾贤志'+99 分)”，返回值为“曾贤志 99 分”，除此之外，加号还可以连接其它对象，后面将逐步讲解。

1.9.2 减 (-)

减运算符也可以执行数字运算，比如代码“`print(100-99)`”，返回值为“1”。

1.9.3 乘 (*)

乘运算符的数字运算，比如代码“`print(100*99)`”，返回值为“9900”。也可以进行字符重复运算，比如代码：“`print('python!'*3)`”，返回值为“python!python!python!”。可以推算出“*3”表示重复3次的意思。除此之外，乘号还可以重复其它对象，后面将逐步讲解。

1.9.4 除 (/)

除运算符的数字运算，比如代码“`print(63/8)`”，返回值为“7.875”。

1.9.5 取模 (%)

取模运算，也就是求余数。比如代码“`print(63%8)`”，返回值为“7”，这个值便是63除8之后的余数。

1.9.6 幂 (**)

返回x的y次幂，比如代码“`print(4**8)`”，返回值为“65536”。

1.9.7 取整数 (//)

取整运算，也就是相除之后商的整数部分。比如代码“`print(63//8)`”，如果直接相除的商为7.875，但只取整数部分，所以结果为“7”。

1.10 比较运算符

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347183>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Python 中的比较运算符有等于(==)、不等于(!=)、大于(>)、小于(<)、大于等于(>=)、小于等于(<=)。

1.10.1 等于 (==)

“==”比较对象是否相等，比如代码“`print(9==9)`”，返回值为True，比如代码“`print(9==8)`”，返回值为False。

1.10.2 不等于 (!=)

“!=” 比较两个对象是否不相等，比如代码 “`print(9!=8)`”，返回逻辑值为 `True`，又比如代码 “`print(9!=9)`”，返回结果为 `False`。

1.10.3 大于 (>)

“>” 返回 `x` 是否大于 `y`。比如代码 “`print(9>8)`”，返回逻辑值为 `True`，又比如代码 “`print(9>9)`”，返回结果为 `False`。

1.10.4 小于 (<)

“<” 返回 `x` 是否小于 `y`。比如代码 “`print(8<9)`”，返回逻辑值为 `True`，又比如代码 “`print(9<9)`”，返回结果为 `False`。

1.10.5 大于等于 (>=)

“>=” 返回 `x` 是否大于等于 `y`。比如代码 “`print(8>=9)`”，返回逻辑值为 `False`，又比如代码 “`print(9>=9)`”，返回结果为 `True`。

1.10.6 小于等于 (<=)

“<=” 返回 `x` 是否小于等于 `y`。比如代码 “`print(8<=9)`”，返回逻辑值为 `True`，又比如代码 “`print(9<=8)`”，返回结果为 `False`。

1.11 赋值运算符

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347182>

全套视频链接：<http://edu.51cto.com/sd/e9508>

赋值运算符是 “=” 号，表示将等号右侧的对象赋予给等号左侧的变量，就比如皇帝赐了一把上方宝剑给你，这里皇帝就是对象，上方宝剑就是变量。上方宝剑本是一把普通的剑，但由于是皇帝给的，那么这把剑就代表了皇帝，你拿到剑后，就可以代表皇帝行使权利。

1.11.1 赋值运算

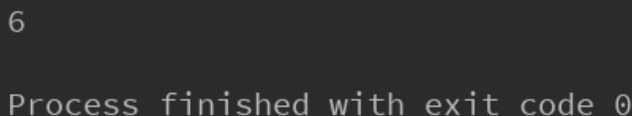
前面我们讲过变量，也介绍了对象。如果要将对象赋值给变量，则使用赋值运算符 “=”。比如 “`n=99`”，表示将对象 99 赋值给变量 `n`，或者理解成将对象 99 贴上标签 `n`，这样 `n` 就代了对象 99。

1.11.2 累积式赋值运算

累积式赋值运算是编程中的一项重要技术。为了让大家更容易明白累积式赋值运算的运算过程，我们先来看如下代码：

```
1-> n=0
2-> n=n+1
3-> n=n+2
4-> n=n+3
5-> print(n)
```

输出结果如图 1.11-1 所示，我们来分析一下上面的代码。



```
6
Process finished with exit code 0
```

图 1.11-1

第 1 句：首先将 0 赋值给变量 n，此时 n 对应的值是 0。

第 2 句：将 n 对应的 0，再加上 1，再将相加结果赋值给 n，此时 n 对应的值是 1。

第 3 句：将 n 对应的 1，再加上 2，再将相加结果赋值给 n，此时 n 对应的值是 3。

第 4 句：将 n 对应的 3，再加上 3，再将相加结果赋值给 n，此时 n 对应的值是 6。

第 5 句：最后使用 print 将 n 对应的值 6 打印输出。

累积式赋值运算，除了像上面的代码写法，也可以简化成如下方式：

```
1-> n=0
2-> n+=1
3-> n+=2
4-> n+=3
5-> print(n)
```

总结一下，上面的代码只是相加的累积式赋值，也可以进行-（减）、*（乘）、/（除）、%（取模）、**（幂）、//（取整数）等累积式的赋值运算。

1.12 逻辑运算符

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347181>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Python 中支持 and（与）、or（或）、not（非）三种逻辑运算符，分别代表三种不同的处理方式。

1.12.1 and（与）

and 运算符，表示多个表达式返回的逻辑值结果必须为 True，则最后返回的结果为 True，否则返回 False。比如代码“print(1>0 and 9>7)”，返回值为 True。代码“print(1>0 and 9>17)”

则返回逻辑值 False。

1.12.2 or（或）

or 运算符，表示多个表达式，返回结果至少有一个逻辑值 True，则最后返回的结果为 True，否则如果所有条件均不成立返回，则返回 False。比如代码“`print(1>0 or 9>7)`”，返回值为 True；代码“`print(1>0 and 9>17)`”则返回逻辑值 True；代码“`print(1>10 and 9>17)`”则返回逻辑值 False。

1.12.3 not（非）

not 运算符，如果表达式返回的逻辑值结果相反。比如代码“`print(not 1>0)`”，本来 1>0 应该返回 True，但是前面加了一个 not，所以返回相反逻辑值 False。

1.13 成员运算符

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347180>

全套视频链接：<http://edu.51cto.com/sd/e9508>

除了前面讲解的算术运算符、比较运算符等运算符之外，Python 还支持成员运算符，成员运算符，是测试字符串、列表等对象中是否包含指定的值。成员运算符是 in。

1.13.1 成员运算符（in）

如果在指定的对象中找到指定的值，则返回 True，否则返回 False。也可以使用 not in 来测试对象中没有指定的值。比如代码“`print('曾贤志' in 'IT 部 曾贤志 2000')`”，在字符串'IT 部 曾贤志 2000'中有包含'曾贤志'，所以最后返回逻辑值为 True。后面还有好多对象都可以使用 in 成员运算符来做测试判断。

1.14 格式化字符串

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347179>

全套视频链接：<http://edu.51cto.com/sd/e9508>

在 python 中会经常格式化字符串，格式化字符串分为百分号(%)方法和 format 方法。不过本节只讲百分号方法，format 函数方法在后面的章节会讲解。

百分号方式格式：`%[(name)][flags][width].[precision]typecode`，接下说明一下这些参数。

[(name)] 可选，用于选择指定的 key

[flags] 可选，可供选择的值有：

+, 右对齐，正数前加正号，负数前加负号。

-, 左对齐，正数前无符号，负数前加负号。

空格，右对齐，正数前加空格，负数前加负号。

0, 右对齐，正数前无符号，负数前加负号，用 0 填充空白处。

[width] 可选，占有宽度。

[precision] 可选，小数点后保留的位数。

typecode 必选

s, 获取传入的对象 __str__ 方法的返回值，并将其格式化到指定位置

d, 将整数、浮点数转化为十进制表示，并将其格式化到指定位置

f, 将整数、浮点数转换成浮点数表示，并将其格式化到指定位置（默认保留小数点后 6 位）


1.15 完美看清代码运行过程

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347178>

全套视频链接：<http://edu.51cto.com/sd/e9508>

在使用 pycharm 编辑代码时，为了了解代码的运行过程。可以通过设置断点和 F8 的方法，看代码是如何一句一句执行的，测试代码如下所示：

```
1-> num=input('请输入分数：')    #输入一个数字
2-> num=float(num)                 #将 num 变量转换为浮点型数字
3-> num/=2                         #将变量 num 获得值除以 2
4-> print('你的最后得分为：%f分'%num)  #输出最后的结果
```

第 1 步：如图 1.15-1 所示，首先在第 1 句代码前单击，出现红圆点，表示在此位置设置断点，也就是当代码运行到这句时就暂停。然后再点击  图标，或者按快捷键 Shift+F9 开始调试。

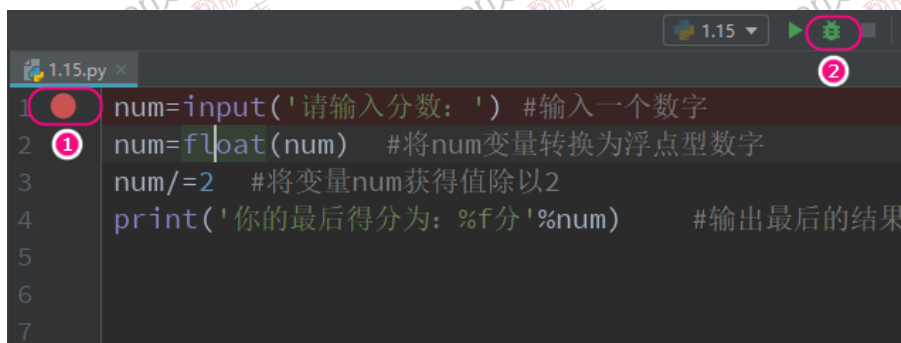


图 1.15-1

第 2 步：在第 1 步，一切准备就绪之后，按功能键 F8 开始单步运行代码。第 1 次按 F8 键效果如图 1.15-2 所示，提示用户输入一个分数。比如输入 99，再按回车键确认。

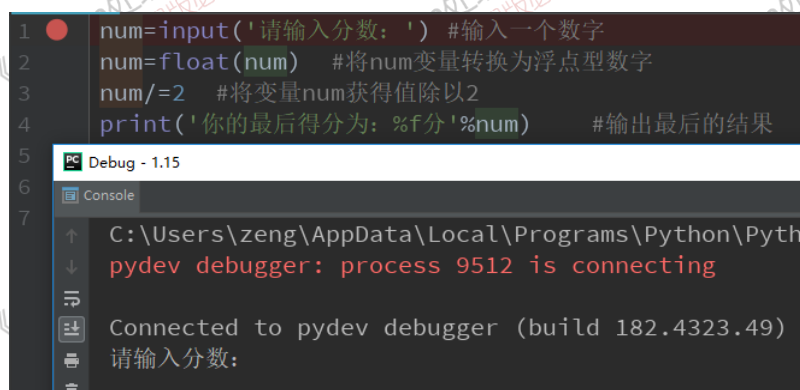


图 1.15-2

此时，在第 1 行代码右侧看到“num:'99'”，表示变量 num 的值是'99'，由于'99'是被包含在单引号中，说明此数字是字符型的。

```
1 num=input('请输入分数: ') #输入一个数字 num: '99'
2 num=float(num) #将num变量转换为浮点型数字
3 num/=2 #将变量num获得值除以2
4 print('你的最后得分为: %f分'%num) #输出最后的结果
```

图 1.15-3

继续第 2 次按 F8 键，此时运行完第 2 句，效果如图 1.15-4 所示，第 1 句后面的 num 变量对应的值变成了 99.0，转换成了浮点型的数字。

```
1 num=input('请输入分数: ') #输入一个数字 num: 99.0
2 num=float(num) #将num变量转换为浮点型数字
3 num/=2 #将变量num获得值除以2
4 print('你的最后得分为: %f分'%num) #输出最后的结果
```

图 1.15-4

继续第 3 次按 F8 键，此时运行完第 3 句，效果如图 1.15-5 所示，第 1 句后面 num 变量对应的值变成了 49.2，如果第 2 句代码不转换，则第 3 句会出错，因为'99'是字符型数字，是不能与数字做算术运算的，类型不相同嘛。

```
1 num=input('请输入分数: ') #输入一个数字 num: 49.5
2 num=float(num) #将num变量转换为浮点型数字
3 num/=2 #将变量num获得值除以2
4 print('你的最后得分为: %f分'%num) #输出最后的结果
5
```

图 1.15-5

继续第 4 次按 F8 键，此时运行完第 4 句，效果如图 1.15-6 所示，此时在屏幕上打印出最后的显示效果。至此运行完所有代码。

```
1 num=input('请输入分数: ') #输入一个数字
2 num=float(num) #将num变量转换为浮点型数字
3 num/=2 #将变量num获得值除以2
4 print('你的最后得分为: %f分'%num) #输出最后的结果
5
Debug - 1.15
Console
C:\Users\zeng\AppData\Local\Programs\Python\Python39\python.exe
pydev debugger: process 29156 is connecting
Connected to pydev debugger (build 182.4323.49)
请输入分数: 99
你的最后得分为: 49.500000分
```

图 1.15-6

如果不再设置断点调试，则再次单击红点取消。对于代码的调试运行，还有很多不同的方式。这里只介绍了最基本的用法，想了解更多关于 pycharm 软件的使用法，可以到 pycharm 的官方网站进行了解。

第2章 用 Python 对 Excel 读写数据

2.1 什么是模块、包、库

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347177>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Module（模块）：扩展名为.py 的文件就可以称做为模块。

Package（包）：在模块之上的概念，为了方便管理.py 模块文件进行打包。包目录下第一个文件必须是 `__init__.py`，否则就是普通的文件夹目录，然后是一些模块文件和子目录，假如子目录中也有 `__init__.py`，那么它就是这个包的子包了。

Library（库）：具有相关功能模块的集合。这也是 Python 的一大特色之一，即具有强大的标准库、第三方库以及自定义模块。

标准库，就是安装 Python 时自带的模块。

第三方库，就是由第三方机构发布的，具有特定功能的模块。用户也可以自行开发模块来供自己和其他人使用。

2.2 安装 Excel 读取库 xlrd

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347176>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Python 标准库中没有直接内置处理 Excel 文件的库，这里我们安装一个老牌的处理 Excel 文件的第三方库 xlrd，操作方法如下：

第 1 步：如图 2.2-1 所示，在 Pycharm 界面中，单击【File】菜单→【Setting...】命令（或者按组合键 `Ctrl+Alt+S`）。

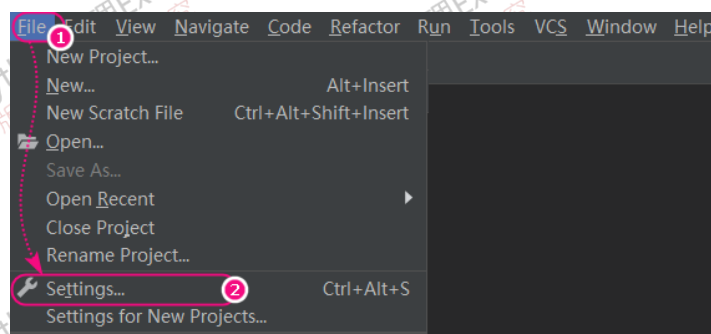


图 2.2-1

第 2 步：如图 2.2-2 所示，在弹出的【Settings】对话框中，单击【Project:project】菜单→【Project Interpreter】命令，然后再单击右侧的■按钮即可。

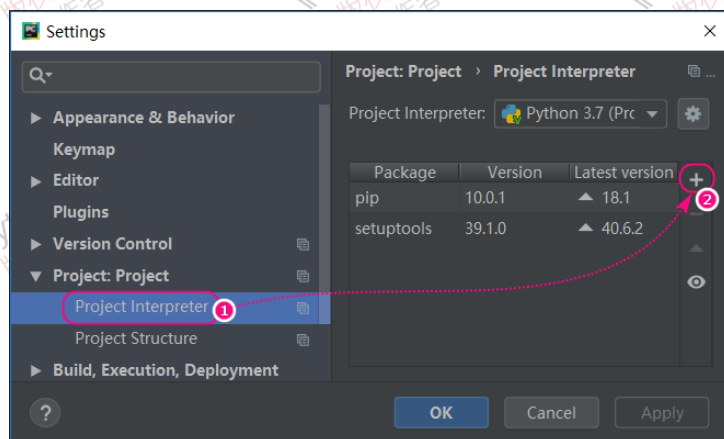


图 2.2-2

第 3 步：如图 2.2-3 所示，在弹出的【Available Packages】对话框，然后输入要搜索的“xlrd”库，然后选择【xlrd】库，之后再单击【Install Package】按钮，开始进行“xlrd”库的安装，安装完成后单击 ×（关闭按钮）。

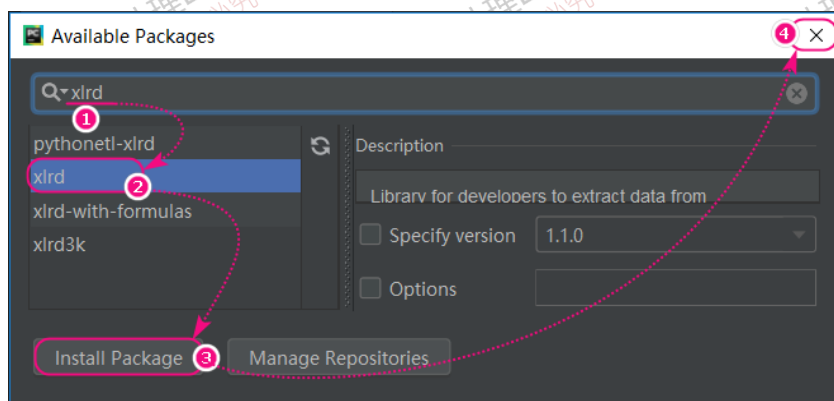


图 2.2-3

第 4 步：如图 2.2-4 所示，有提示“Package 'xlrd' installed successfully”，也可以看到【xlrd】库的相关版本信息，证明已经安装成功。

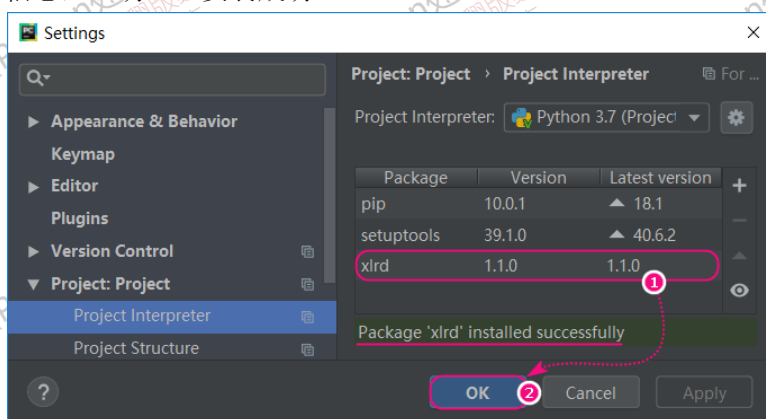


图 2.2-4

第 5 步：如图 2.2-5 所示，可以在 Pycharm 界面看到 xlrd 库的位置，也可以看到已经对应安装到了“D:\Project\venv\Lib\site-packages”文件夹中。



图 2.2-5

2.3 xlrd 模块导入

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347175>

全套视频链接：<http://edu.51cto.com/sd/e9508>

导入模块的格式为：`import 模块名`

比如导入 xlrd 写法为：“`import xlrd`”。也可以使用别名命名方式，如：“`import xlrd as xl`”，后续就可以直接使用 `xl` 来代表 xlrd 模块。

还有一种调用格式为：`from 模块名 import 方法名`

比如导入 xlrd 模块中的 `open_workbook`，写法为：“`from xlrd import open_workbook`”，也可以使用别名命名方式，如：“`from xlrd import open_workbook as owb`”，后续就可以直接使用 `owb` 来代表 `open_workbook` 工作簿对象了。

注意，xlrd 只能读取 xls 格式工作簿数据。

2.4 读取 Excel 工作簿、工作表信息

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347174>

全套视频链接：<http://edu.51cto.com/sd/e9508>

在安装读取 xls 文件的库之后，则需要读取 Excel 文件的相关信息，本节学习一些常见的获取工作簿、工作表对象的方法。

```
1-> import xlrd    #导入xlrd 模块
2-> wb=xlrd.open_workbook('工作簿名.xls')    #读取工作簿
3-> wsobj=wb.sheets()    #读取工作簿下的所有工作表对象
4-> wsname=wb.sheet_names()    #读取工作簿下的所有工作表名称
5-> ws1=wb.sheet_by_name('工作表名')    #按指定名称读取工作表对象-方法1
6-> ws2=wb.sheet_by_index(1)    #按指定序号读取工作表对象-方法2
7-> ws3=wb.sheets()[1]    #按指定序号读取工作表对象-方法3
```

2.5 读取 Excel 行、列、单元格信息

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347173>

全套视频链接：<http://edu.51cto.com/sd/e9508>

前面学习了如何获取 Excel 工作簿、工作表的信息，应该继续细化下去，本节获取行、列、单元格的信息。

```
1-> import xlrd
2-> ws=xlrd.open_workbook('工作簿名.xls').sheet_by_name('工作表名')    #获取工
    作表对象
3-> crow=ws.nrows    #获取行数
4-> ccol=ws.ncols    #获取列数
5-> row_data=ws.row_values(行号)    #获取指定行数据
6-> col_data=ws.col_values(列号)    #获取指定列数据
7-> cell_data_1=ws.cell_value(行号,列号)    #获取单元格数据
8-> cell_data_2=ws.cell(行号,列号).value    #获取单元格数据
```

注意，如果以序号方式获取工作表、单元格的数据，均是从序号 0 开始的。

2.6 安装 Excel 写入库 xlwt

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347172>

全套视频链接：<http://edu.51cto.com/sd/e9508>

在 2.2 节学习是 xlrd 库的安装，xlrd 只能读取 Excel 文件，并不具有写入的功能，需要使用安装 xlwt 库，这个库才具有创建工作簿、工作表，将数据写入到单元格的功能。

第 1 步：如图 2.6-1 所示，单击【File】菜单→【Settings…】命令，弹出【Settings】对话框，然后单击【Project:Project】下的【Project Interpreter】命令，再单击+按钮。

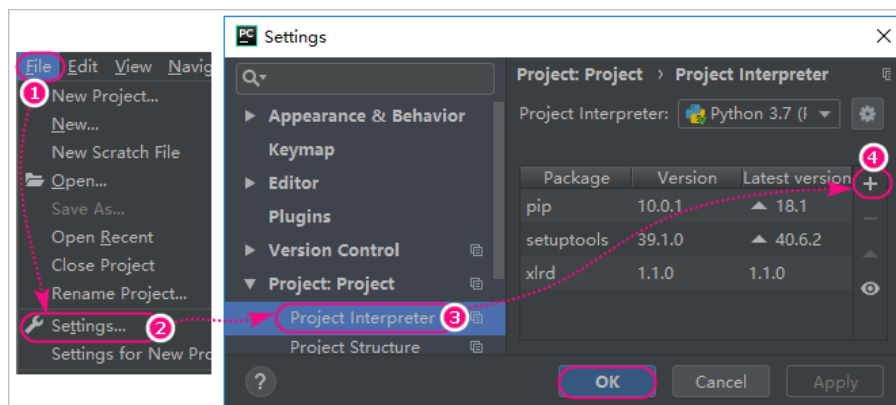


图 2.6-1

第 2 步：如图 2.6-2 所示，在【Available Packages】对话框中搜索“xlwt”，然后单击选择【xlwt】库，再单击【Install Package】按钮，开始安装。安装完成之后再单击×（关闭按钮）。

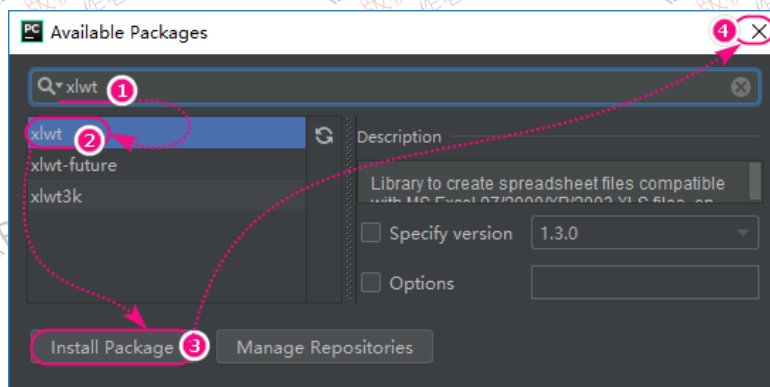


图 2.6-2

第 3 步：如图 2.6-3 所示，回到【Setting】对话框，可以看到已经有了【xlwt】库，最后再单击【Ok】按钮完成安装。

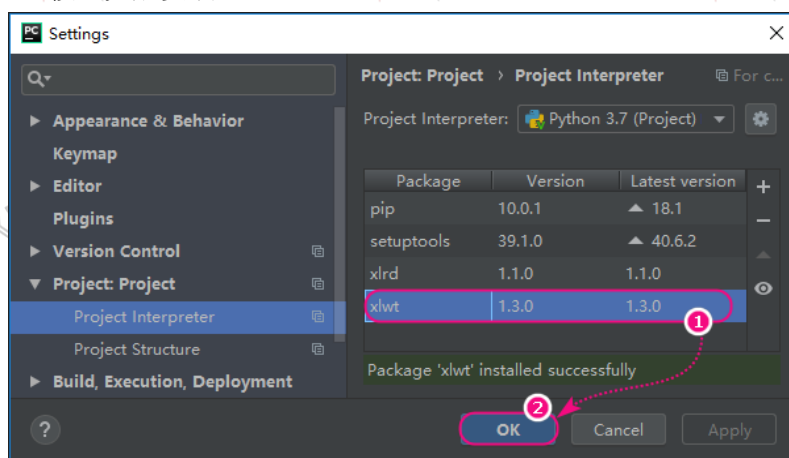


图 2.6-3

2.7 创建工作簿、工作表和写入单元格

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347171>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Xlwt 可以创建工作簿、保存工作簿。创建工作表，以及将数据写单元格等常见操作。

```
1-> import xlwt    #引入 xlwt 模块
2-> nwb=xlwt.Workbook(encoding='utf-8') #新建工作簿
3-> nws=nwb.add_sheet('工作表名')    #添加工作表
4-> nws.write(行号,列号,'要写入的值') #写入单元格
5-> nwb.save('工作簿名.xls') #保存工作簿
```


2.8 安装 Excel 修改库

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347170>

全套视频链接：<http://edu.51cto.com/sd/e9508>

前面，学习了利用 xlrd 库读取工作簿、工作表、单元格等相关信息，也学习了利用 xlwt 库新建工作簿、工作表、将数据写入单元格等技术。如果我们需要将对打开的工作簿进行修改又该怎么做呢？xlrd 和 xlwt 各自都无法独立完成，但只要在他们之间建立一个桥梁，就可以实现打开修改的功能。

接下来安装起到桥梁作用的 xlutils 库，但要注意，要使用 xlutils 库就一定要安装 xlrd 和 xlwt 库。xlutils 库的安装方法如下：

第 1 步：如图 2.8-1 所示，已经有安装 xlrd 和 xlwt 库了，接下来单击【File】菜单→【Settings...】命令，弹出【Settings】对话框，然后单击【Project:Project】下的【Project Interpreter】命令，再单击  按钮。

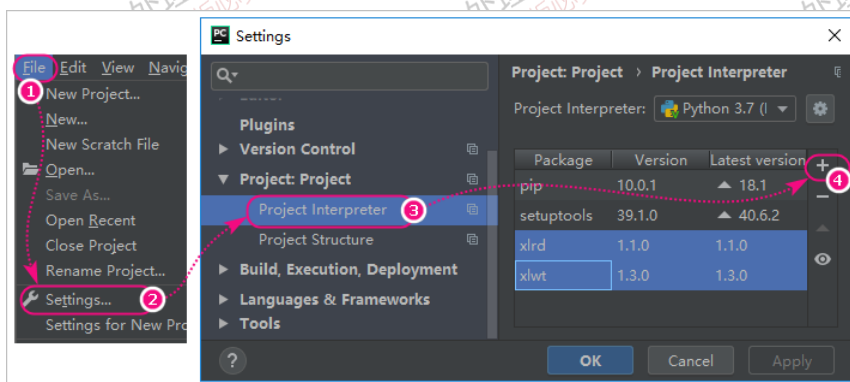


图 2.8-1

第 2 步：如图 2.8-2 所示，在【Available Packages】对话框中搜索“xlu”，即可看到 xlutils 库，单击选择【xlutils】，再单击【Install Package】按钮，开始安装。安装完成之后再单击 ×（关闭按钮）。

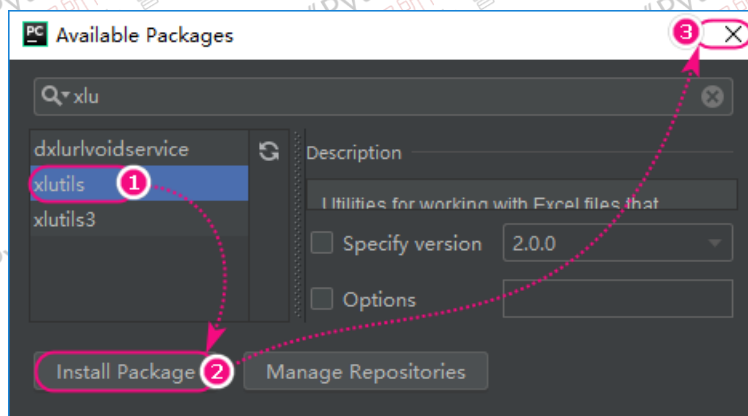


图 2.8-2

第 3 步：如图 2.8-3 所示，回到【Setting】对话框，可以看到已经有了【xlutils】库，这时 xlrd、xlwt、xlutils 三个库都同时存在了，最后再单击【OK】按钮完成安装。

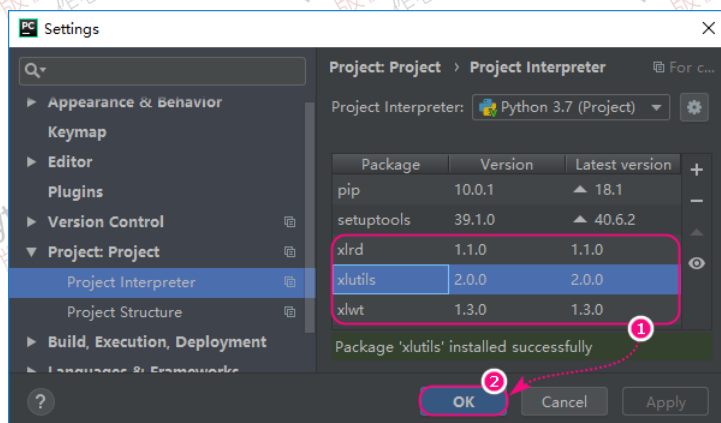


图 2.8-3

2.9 修改工作簿、工作表、单元格

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347169>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd    #导入 xlrd 模块
2-> from xlutils.copy import copy    #导入 xutils 中的复制模块
3-> wb=xlrd.open_workbook('工作簿名.xls')    #读取工修改的工作簿
4-> nwb=copy(wb)    #复制工作簿
5-> nws1=nwb.add_sheet('工作表名')    #新建工作表
6-> nws2=nwb.get_sheet(序号)    #读取工作表
7-> nws3=nwb.get_sheet('工作表名')    #读取工作表
8-> nws1.write(行号,列表,'值')    #将数据 写入指定工作表
9-> nwb.save('工作簿名.xls')    #保存工作簿
```

注意，最后 `nwb.save` 保存的工作簿可以和打开时的工作簿名一样，也可以不一样，如果一样，则是覆盖原工作簿，不一样则是另存工作簿。
还有，在运行代码时，被修改的工作簿必须关闭。

第3章 循环语句与分支语句

循环语句和条件分支语句在任何编程语言中都是必学的知识，此两大点知识是批量处理数据的核心。

3.1 for...in 循环语句基础

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347168>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Python for 循环可以遍历任何序列的项目，如一个字符串或者一个列表。如图 3.1-1 所示，列出了 for 循环的语法结构。



图 3.1-1

注意，循环处理语句必须在 for 循环语句下，也就是 for 下面运行的语句必须要缩进。否则，会提示错误。

3.1.1 循环字符串

for 循环语句除了遍历指定范围的序列值，也可以遍历指定的字符串，将字符串中的每个字符逐个输出。

```
1-> for s in 'python':
2->     print(s)
```

返回结果如图 3.1-2 所示。

```
p
y
t
h
o
n

Process finished with exit code 0
```

图 3.1-2

其实，for 循环语句除可以遍历指定范围的数字、字符串之外，还可以遍历列表、元素、集合、字典等对象。

3.1.2 循环指定范围序列数

如果我们要循环指定范围的数据，可以使用 `range` 函数，一般通过使用 `for` 循环语句来逐个取得数字。

`range` 函数语法为：`range(start, stop[, step])`，参数说明如下：

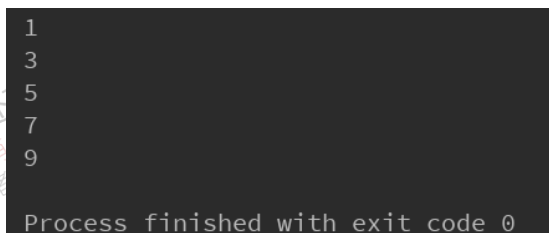
start：计数从 `start` 开始。默认是从 0 开始。

stop：计数到 `stop` 结束，但不包括 `stop`。

[step]：步长，默认为 1

```
1-> for n in range(1,10,2):
2->     print(n)
```

返回结果如图 3.1-3 所示，表示从 1 开始，循环遍历到 10（10 不包含在内）的序列值，序列差值为 2。



```
1
3
5
7
9
Process finished with exit code 0
```

图 3.1-3

3.2 for...in 循环语句应用（批量创建工作簿）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347167>

全套视频链接：<http://edu.51cto.com/sd/e9508>

实例应用：使用 `for` 循环语句创建 1 到 12 月的工作簿，每个工作簿中创建一个名为统计表的工作表，代码如下所示：

```
1-> import xlwt    #导入 Excel 写入模块
2-> for m in range(1,13):    #循环 1~13 的循环序列数
3->     wb=xlwt.Workbook(encoding='utf-8')    #创建工作簿
4->     wb.add_sheet('统计表')    #创建工作表（工作簿中至少要有有一个工作表）
5->     wb.save('%02d 月.xls'%m)    #保存工簿
```

3.3 for...in 嵌套循环语句

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347166>

全套视频链接：<http://edu.51cto.com/sd/e9508>

`for` 循环语句可以嵌套使用，但每一层 `for` 循环语句注意缩进，下面使用 `for` 嵌套循环方式完成九九乘法表的制作。

```
1-> for x in range(1,10):
2->     for y in range(1,x+1):
3->         print('%d×%d=%-5d'%(y,x,y*x),end="")
4->     print("")
```

3.4 for...in 嵌套循环语句应用（制作九九乘法表）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347165>

全套视频链接：<http://edu.51cto.com/sd/e9508>

实例：将九九乘法表写入 Excel 表格中，只需要定位好对应的单元格，然后将格式化好的数据写入单元格即可。

```
1-> import xlwt #导入写入模块
2-> wb=xlwt.Workbook(encoding='utf-8') #创建工作簿
3-> ws=wb.add_sheet('乘法表') #创建工作表
4-> for x in range(1,10): #外层for 循环
5->     for y in range(1,x+1): #内层for 循环
6->         ws.write(x,y,'%d×%d=%d'%(x,y,x*y)) #将乘法表写入单元格
7-> wb.save('乘法表.xls') #保存工作簿
```

3.5 while 循环语句

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347164>

全套视频链接：<http://edu.51cto.com/sd/e9508>

while 循环是当 while 后面的条件(表达式)为真,才执行 while 循环体内的 do something,直到条件为假时,退出循环。

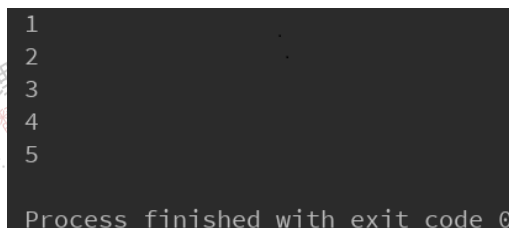


图 3.5-1

注意，循环处理语句必须在 while 循环语句下，也就是 while 下面运行的语句必须要缩进。否则，会提示错误。

```
1-> n=0 #初始化变量n
2-> while n<5: #循环条件为当n 小于5 时
3->     n+=1 #累加数字
4->     print(n) #在屏幕打印n 值
```

返回结果如图 3.5-2 所示，生成的序列数差值为 1，序列差值取决于 $n+=1$ ，比如希望差值为 2，则语句为： $n+=2$ ，其它差值以此类推。



```
1
2
3
4
5
Process finished with exit code 0
```

图 3.5-2

3.6 while 循环语句应用（读取工作簿信息到新表）

本门课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347163>

全套视频链接：<http://edu.51cto.com/sd/e9508>

实例：将“招生表.xls”工作簿下的所有工作表名称写入“统计结果.xls”工作簿下的“统计表”。并且给每个工作表编写一个序号。

```
1-> import xlrd,xlwt #导入读取、写入模块
2-> wb=xlrd.open_workbook('招生表.xls') #打开工作簿
3-> nwb=xlwt.Workbook(encoding='uft-8') #新建工作簿
4-> nws=nwb.add_sheet('统计表') #新建工作表
5-> n=0 #初始化变量0
6-> while n<wb.sheets().__len__(): #当n 小于工作表个数则循环
7->     nws.write(n,0,'表%d'%n) #将序号写入第1 列
8->     nws.write(n,1,wb.sheets()[n].name) #将工作簿下表名写入第2 列
9->     n+=1 #累加变量n
10-> nwb.save('统计结果.xls') #保存新建的工作簿
```

3.7 while 嵌套循环语句

本门课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347162>

全套视频链接：<http://edu.51cto.com/sd/e9508>

使用 while 语句制作九九乘法表，同 for 语句一样，同样需要双循环，思路基本相似。


```
1-> x,y=0,0 #初始化变量x,y
2-> while x<9: #当x 小于9, 则执行循环
3->     x+=1 #累加变量x 的值为1
4->     while y<x: #y 小于x 时, 则执行循环
5->         y+=1 #累加变量y 的值为1
6->         print('%d×%d=%02d\t'%(y,x,x*y),end=") #格式化九九乘法表
7->     y=0 #将内层变量初始化为0, 方便下一次内层循环又从0 开始
8->     print("") #换行
```

3.8 while 嵌套循环语句应用(批量创建工作簿、工作表)

本节课程链接: <http://edu.51cto.com/center/course/lesson/index?id=347161>

全套视频链接: <http://edu.51cto.com/sd/e9508>

实例: 新建 2015~2018 年的工作簿文件, 每个工作簿文件中再分别新建 1~12 月的工作表。本实例使用 while 双循环完成。

```
1-> import xlwt #导入 Excel 的写入模块
2-> x,y=2014,0 #初始化变量x,y
3-> while x<2018: #如果x 小于2018, 则循环
4->     wb=xlwt.Workbook(encoding='uft-8') #创建工作簿
5->     x+=1#累加变量x
6->     while y<12: #如果y 小于12, 则循环。
7->         y+=1#累加变量y
8->         ws=wb.add_sheet('%d 年%d 月'%(x,y)) #创建工作表
9->         y=0 #初始化变量y 为0
10->     wb.save('%d 年.xls'%x) #保存工作簿
```

3.9 if 条件语句

本节课程链接: <http://edu.51cto.com/center/course/lesson/index?id=347160>

全套视频链接: <http://edu.51cto.com/sd/e9508>

if 条件分支语句, 是根据条件测试的结果决定不同的处理方向。语法结构如图 3.9-1 所示, if 关键字在最前面, 然后紧跟条件测试表达式, 再写上冒号, 表示条件语句结束, 换行到下一行, 注意, 此时一定要缩进, 然后再编写当条件成立时的处理语句。如果条件不成立则忽略不处理。



图 3.9-1

```
1-> score=input('请输入分数: ')    #输入要判断分数
2-> score=int(score)    #将字符型数字转换成整数
3-> if score>90:    #判断分数是否大于90 分
4->     print('优秀')    #如果条件成立，则返回优秀
```

3.10 if…else 条件语句

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347159>

全套视频链接：<http://edu.51cto.com/sd/e9508>

比如判断如果输入的分数大于等于 90 分，则返回“优秀”，否则返回“普通”。

```
1-> score=input('请输入分数: ')    #输入要判断分数
2-> score=int(score)    #将字符型数字转换成整数
3-> if score>90:    #判断分数是否大于90 分
4->     print('优秀')    #如果条件成立，则返回优秀
5-> else:
6->     print('普通')    #如果条件成立，则返回普通
```

3.11 多条件 if 语句 1（将筛选结果写入新工作簿）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347158>

全套视频链接：<http://edu.51cto.com/sd/e9508>

判断语文和数学分数是否同时大于等于 90，如果条件成立则返回优秀，否则返回普通。

```
1-> chinese=input('请输入语文分数: ')    #输入语文分数
2-> maths=input('请输入数学分数: ')    #输入数学分数
3-> if int(chinese)>=90 and int(maths)>=90:    #如果两科成绩同时大于等于90
4->     print('优秀')    #条件成立则返回优秀
5-> else:    #否则
6->     print('普通')    #条件不成立则返回普通
```

实例应用：将指定工作簿指定工作表下语文、数学两列成绩同时大于等于 80 的记录写

入新工作簿。

```
1-> import xlrd,xlwt    #导入读取、写入模块
2-> wb=xlrd.open_workbook('成绩表.xls') #读取工作簿文件
3-> ws=wb.sheet_by_name('分数表') #读取工作表
4-> nwb=xlwt.Workbook(encoding='uft-8') #创建工作簿
5-> nws=nwb.add_sheet('结果') #创建工作表
6-> n,m=0,0 #初始化变量n,m
7-> while n<ws.nrows-1: #开始循环数据
8->     n+=1 #累加n 变量做为读取单元格值时的行号
9->     if ws.cell_value(n,1)>=80 and ws.cell_value(n,2)>=80: #判断两科成绩是否大
        于等于80分
10->         m+=1 #累加m 变量，做为条件成立时的写入的行号
11->         nws.write(m,0,ws.cell_value(n,0)) #写入姓名
12->         nws.write(m,1,ws.cell_value(n,1)) #写入语文成绩
13->         nws.write(m,2,ws.cell_value(n,2)) #写入数学成绩
14-> nws.write(0,0,'姓名') #写入标题
15-> nws.write(0,1,'语文') #写入标题
16-> nws.write(0,2,'数学') #写入标题
17-> nwb.save('筛选结果.xls') #保存工作簿
```

3.12 多条件 if 语句 2（根据总分判断等级）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347157>

全套视频链接：<http://edu.51cto.com/sd/e9508>

If 语句除了同时判断多个条件，也可以递次判断多个条件。语法结构为：

```
if conditional_test:
    do something
elif conditional_test:
    .....
else:
    do something
```

实例应用：判断语文和数学两个科目之和是属于哪个级别。

```
1-> import xlrd    #导入读取模块
2-> from xlutils.copy import copy #导入复制模块
3-> wb=xlrd.open_workbook('成绩表.xls') #读取工作簿
4-> ws=wb.sheet_by_name('分数表') #读取工作表
5-> nwb=copy(wb)    #复制读取工作簿
6-> nws=nwb.get_sheet('分数表')    #读取工作表
7-> n=0
8-> while n<ws.nrows-1:
9->     n+=1
10->     subtotal=ws.cell_value(n,1)+ws.cell_value(n,2) #将语文和数学相加
11->     if subtotal>=180: #如果大于等于 180
12->         nws.write(n,3,'优') #则写入优
13->     elif subtotal>=160: #如果大于等于 160
14->         nws.write(n,3,'良')    #则写入良
15->     elif subtotal>=120: #如果大于等于 120
16->         nws.write(n,3,'中')    #则写入中
17->     else:
18->         nws.write(n,3,'差')    #则写入差
19-> nwb.save('成绩表.xls') #保存工作簿
```

3.13 break 语句（跳出整个循环）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347156>

全套视频链接：<http://edu.51cto.com/sd/e9508>

终止 break 语句所在的循环，当循环或判断执行到 break 语句时,即使判断条件为 True 或者序列尚未完全被历遍,都会跳出循环或判断。

for 循环语句中的 break 应用

```
1-> for c in 'abcdefg': #遍历字符串
2->     if c=='d': #如果遍历的字符串等于 d
3->         break    #则跳出当前 for 循环
4->     else:
5->         print(c) #在屏幕上打印出字符
```

while 循环语句中的 break 应用

```
1-> n=0 #初始化变量n
2-> while True:
3->     n+=1 #累加变量n
4->     if n==5: #如n 等于5
5->         Break #则跳出循环
6->     else:
7->         print(n) #打印出变量n
```

3.14 continue 语句（跳出当次循环）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347155>

全套视频链接：<http://edu.51cto.com/sd/e9508>

当循环或判断执行到 continue 语句时,continue 后的语句将不再执行,会跳出当次循环,继续执行循环中的下一次循环。

for 循环语句中的 continue 应用

```
1-> for c in range(1,11):#遍历1~11 之间数字
2->     if c==5:#如果变量n 等于5
3->         print('我是五')#返回指定值
4->         continue#跳出当次循环，不再执行此句之后的语句，继续下一次循环
5->     print(c)
```

while 循环语句中的 continue 应用

```
1-> n=0#初始化变量n
2-> while n<6:#如果变量n 小于6
3->     n+=1#累加变量n
4->     if n==3:#当变量n 等于3
5->         print('我是三')#返回指定值
10->         continue#跳出当次循环，不再执行此句之后的语句，继续下一次循环
11->     print(n)
```

3.15 综合应用

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347154>

全套视频链接：<http://edu.51cto.com/sd/e9508>

实例应用：要求统计出业绩表中每个人的累计业绩大于等于 1000 则表示达标，需要统计出开始达标的月份以及业绩。


```
12-> import xlrd      #导入读取模块
13-> from xlutils.copy import copy      #导入复制模块
14-> wb=xlrd.open_workbook('业绩表.xls') #读取工作簿
15-> ws=wb.sheet_by_name('Sheet1') #读取 Sheet1 工作表
16-> nwb=copy(wb)      #复制工作簿
17-> nws=nwb.get_sheet('统计结果') #读取工作簿下的统计结果工作表
18-> subtotal,n=0,0     #初始化变量 subtotal、n
19-> for r in range(1,ws.nrows): #遍历行号
20->     for c in range(1,13):    #遍历列号
21->         subtotal+=ws.cell_value(r,c) #累加业绩数据
22->         if subtotal>=1000: #如果大于等于 1000
23->             n+=1      #累加变量
24->             nws.write(n,0,ws.cell_value(n,0)) #将姓名写入单元格
25->             nws.write(n,1,'%d 月份'%c) #将月份写入单元格
26->             nws.write(n,2,subtotal) #将业绩写入单元格
27->             Break     #退出当前循环
28->     subtotal=0      #重置变量
29-> nwb.save('业绩表.xls') #保存工作簿
```

第4章 字符串处理技术

4.1 字符串切片

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347153>

全套视频链接：<http://edu.51cto.com/sd/e9508>

字符串是最常用的一种数据类型，Python 中的字符串是不能修改的，只能做提取、拆分、合并重组等操作。

4.1.1 提取单个字符

Python 中，提取字符串中指定的位置的单个字符，表示方式为：`str[位置]`，既可以从开头计算，也可以从结尾计算。

从开头提取单个字符的表示方法：

```
1-> s='Excel 我很忙' #提供要被提取的字符串
2-> print(s[0],s[1],s[2]) #分别变量s中的第0、1、2个字符，返回值为：Exc
```

通过上面的案例可以看出是从开头提取字符，以用数字序号来代表位置，而位置是从0开始算起的。

从结尾提取单个字符的表示方法：

```
1-> s='Excel 我很忙'#提供要被提取的字符串
2-> print(s[-1],s[-2],s[-3]) #分别变量s中的第-1、-2、-3个字符，返回值为：忙很我
```

通过上面的案例可以看出是从结尾提取字符，以用数字序号来代表位置，而位置是从-1开始算起的。

4.1.2 提取多个字符

在 4.1.1 节中讲解了提取单个字符的表示方法，如果要提取多个字符呢？它的表示方法则更多了。提取字符串中某一段字符的表示方法为：`str[开始位置:结束位置]`

```
s='Excel 我很忙'
1-> #从左计算：指定的起始位置到指定的终止位置
2-> print(s[2:5]) #返回值为：cel
3->
4-> #从左计算：从开头提取到指定位置
5-> print(s[:3]) #返回值为：Exc
6->
7-> #从左计算：从指定的起始位置到结束。
8-> print(s[3:]) #返回值为：el 我很忙
9->
10-> #从右计算：指定的起始位置到指定的终止位置
11-> print(s[-3:-1]) #返回值为：我很
12->
13-> #从右计算：指定的起始位置到结束
14-> print(s[-3:]) #返回值为：我很忙
15->
16-> #提取所有字符
17-> print(s[:]) #返回值为：Excel 我很忙
```

字符串提取还可以设置步长值，表达式方式为：str [开始位置:结束位置:步长值]

```
1-> #从左计算，隔一个字符提取一个字符
2-> print(s[::2]) #返回结果为：Ecl 很
3->
4-> #从右计算，反转字符串
5-> print(s[::-1]) #返回结果为：忙很我lecxE
```

4.2 字符串切片应用（整理工作表数据）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347152>

全套视频链接：<http://edu.51cto.com/sd/e9508>

将成绩表工作簿下分数表中的第 1 列中的部门和姓名分别提取到新工作簿中不同的列。

```
1-> import xlrd,xlwt #导入读取、写入模块
2-> wb=xlrd.open_workbook('成绩表.xls') #读取工作簿
3-> ws=wb.sheet_by_name('分数表') #读取工作表
4-> nwb=xlwt.Workbook(encoding='uft-8') #新建工作簿
5-> nws=nwb.add_sheet('Sheet1') #新建工作表
6-> n=0 #初始化变量n
7-> for c in ws.col_values(0): #循环分数表中第1列的数据
8->     nws.write(n,0,c[3]) #提取第1列中的部门数据，并写入。
9->     nws.write(n,1,c[4:]) #提取第1列中的姓名数据，并写入。
10->     nws.write(n,2,ws.cell_value(n,1)) #写入第2列的数据
11->     n+=1 #累加数据
12-> nwb.save('结果表.xls') #保存新建的工作簿
```

4.3 字符串长度（个数）统计

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347151>

全套视频链接：<http://edu.51cto.com/sd/e9508>

4.3.1 len

len 方法返回对象（字符、列表、元组等）长度或项目个数。比如代码：`print(len('Excel'))`，最后返回值为 5。

4.3.2 count

count 方法用于统计字符串里指定字符串出现的次数。可选参数为在字符串搜索的开始与结束位置，语法格式为：`str.count(sub, start= 0,end=len(string))`。

str 被搜索的父字符串
sub 搜索的子字符串
start 字符串开始搜索的位置。默认为第一个字符,第一个字符索引值为 0。
end 字符串中结束搜索的位置。字符串中第一个字符的索引为 0。默认为字符串的最后一个位置。

```
1-> s='销售部-张三，财务部-李四，销售部-王二麻，销售部-林林'
2-> print(s.count('销售部'))#统计销售部在字符串出现的次数。
3-> print(s.count('销售部',5))#指定搜索字符串的起始位置。
4-> print(s.count('销售部',5,18))#指定搜索字符串的起始和结束位置。
```

实例应用：统计“优良中差”4 个等级出现的次数。

```
1-> import xlrd    #导入读取模块
2-> wb=xlrd.open_workbook('等级表.xls') #读取工作簿
3-> ws=wb.sheet_by_name('Sheet1') #读取工作表
4-> n,m=0,0 #初始化变量n、m
5-> for l in '优良中差': #循环等级
6->     while n<ws.nrows-1: #循环行号
7->         n+=1 #累加数据
8->         m +=ws.cell_value(n,1).count(l) #累加等级次数
9->     print(m) #打印等级出现地次数
10->     n,m=0,0 #初始化变量n、m
```

4.4 字符串的查找

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347150>

全套视频链接：<http://edu.51cto.com/sd/e9508>

4.4.1 index

index 方法用于从字符串中找出某个对象第一个匹配项的索引位置，如果查找的字符串不存在会报一个异常。

str.index(sub[,start=0[,end=len(str)]])

str 父字符串

sub 指定检索的子字符串

start 可选参数，开始索引，默认为 0。（可单独指定）

end 可选参数，结束索引，默认为字符串的长度。（不能单独指定）

```
1-> S='A 组 89, B 组 98, C 组 100, D 组 78, B 组 74'
2-> print(S.index('B 组')) #返回每一个匹配项的索引位置
3-> print(S.index('B 组',6)) #指定查找的起始位置
4-> print(S.index('B 组',6,15)) #指定查找的起始位置和终止位置
```

4.4.2 find

find 方法从字符串中找出某个子字符串第一个匹配项的索引位置，该方法与 index 方法一样，只不过如果子字符串不在字符串中不会报异常，而是返回-1。

实例应用：统计等级“优”在下半年数量。


```
1-> import xlrd    #导入读取模块
2-> from xlutils.copy import copy #导入修改模块
3-> wb=xlrd.open_workbook('等级表.xls') #读取工作簿数据
4-> ws=wb.sheet_by_name('Sheet1') #读取工作表数据
5-> nwb=copy(wb)    #复制工作簿
6-> nws=nwb.get_sheet('Sheet1')    #读取工作表数据
7-> n=0 #初始化变量n
8-> while n<ws.nrows-1:    #循环要遍历单元格的行号
9->     n+=1    #累加变量n
10->     s=ws.cell_value(n,1)    #获取要处理的单元格
11->     nws.write(n,2,s.count('优',s.find('\n'))) #将统计结果写单元格
12-> nwb.save('等级表.xls') #保存工作簿
```

4.5 字符串的替换

本节课链接：<http://edu.51cto.com/center/course/lesson/index?id=347149>

全套视频链接：<http://edu.51cto.com/sd/e9508>

replace 方法用于把字符串中指定的旧子字符串替换成指定的新子字符串，如果指定count 可选参数则替换指定的次数，默认全部替换。

str.replace(old,new[,count=str.count(old)])

old 指定的旧子字符串

new 指定的新子字符串

count 可选参数，替换的次数，默认为指定的旧子字符串在字符串中出现的总次数。

```
1-> S='A 组-优秀； B 组-良好； C 组-优秀； D 组-优秀'
2-> print(S.replace('优秀','（晋级）'))
3-> #返回值为： A 组-（晋级）； B 组-良好； C 组-（晋级）； D 组-（晋级）
4-> print(S.replace('优秀','（晋级）',1))
5-> #返回值为： A 组-（晋级）； B 组-良好； C 组-优秀； D 组-优秀
6-> print(S.replace('优秀','（晋级）',2))
7-> #返回值为： A 组-（晋级）； B 组-良好； C 组-（晋级）； D 组-优秀
```

实例应用：将编号两侧的横线用括号替换。

```
1-> import xlrd    #导入读取模块
2-> from xlutils.copy import copy #导入复制模块
3-> wb=xlrd.open_workbook('等级表.xls') #读取工作簿
4-> ws=wb.sheet_by_name('Sheet1') #读取工作表
5-> nwb=copy(wb)    #复制工作簿
6-> nws=nwb.get_sheet('Sheet1')    #读取工作表
7-> n=0 #初始化变量n
8-> while n<ws.nrows-1:    #循环行号
9->     n+=1    #累加变量n
10->     s=ws.cell_value(n,0)    #读取单元格信息
11->     nws.write(n,1,s.replace('-',',').replace('-',','),1)) #将处理后结果写入单元格
12-> nwb.save('等级表.xls ') #保存工作簿
```

4.6 字符串的拆分与合并

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347148>

全套视频链接：<http://edu.51cto.com/sd/e9508>

4.6.1 split

split 方法拆分字符串。通过指定分隔符对字符串进行切片，并返回分割后的字符串列表 (list)，语法结构：str.split(str="",num=string.count(str))。

str 表示为分隔符，默认为空格，但是不能为空("")。若字符串中没有分隔符，则把整个字符串作为列表的一个元素。

num 表示分割次数。如果存在参数 num，则仅分隔成 num+1 个子字符串，并且每一个子字符串可以赋给新的变量。

```
1-> s='张三、李 四、王二 麻子、小明'
2-> print(s.split()) #返回值为: ['张三、李', '四、王二', '麻子、小明']
3-> print(s.split(',')) #返回值为: ['张三', '李四', '王二麻子', '小明']
4-> print(s.split('-', 2)) #返回值为: ['张三', '李四', '王二麻子、小明']
```

4.6.2 join

join() 方法用于将序列中的元素以指定的字符连接生成一个新的字符串。

语法结构：str.join(sequence)

str 分隔符。可以为空

sequence 要连接的元素序列、字符串、元组、字典

```
30-> l=['a','b','c']  
31-> print('.'.join(l)) #返回值: a-b-c
```

第5章 列表处理技术

5.1 列表基础

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347147>

全套视频链接：<http://edu.51cto.com/sd/e9508>

列表是 python 中最常用的对象，熟练掌握列表的各种操作非常关键。列表的格式为[元素 1,元素 2,元素 3……]

列表中元素的数据类型，可以是任意数据类型，但要注意正确的表达方式。

列表中可以再嵌套列表

列表的循环遍历

空列表的表示方法

5.2 列表切片

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347146>

全套视频链接：<http://edu.51cto.com/sd/e9508>

Python 中符合序列的有序序列都支持切片，例如列表，字符串，元组。

切片格式为：[start:end:step]

start 起始索引，从 0 开始，-1 表示结束。

end 结束索引。

step 步长，步长为正时，从左向右取值。步长为负时，反向取值。

注意切片的结果不包含结束索引，即不包含最后的一位，-1 代表列表的最后一个位置索引。

```
1-> l=['a','b','c','d','e','f','g']
2-> print(l[2:6])    #返回: ['c', 'd', 'e', 'f']
3-> print(l[:4])     #返回: ['a', 'b', 'c', 'd']
4-> print(l[5:])     #返回: ['f', 'g']
5-> print(l[-5:-2])  #返回: ['c', 'd', 'e']
6-> print(l[-3:])    #返回: ['e', 'f', 'g']
7-> print(l[-6:6])   #返回: ['b', 'c', 'd', 'e', 'f']
8-> print(l[2:-1])   #返回: ['c', 'd', 'e', 'f']
9-> print(l[:])      #返回: ['a', 'b', 'c', 'd', 'e', 'f', 'g']
10-> print(l[:2])    #返回: ['a', 'c', 'e', 'g']
11-> print(l[::-1])  #返回: ['g', 'f', 'e', 'd', 'c', 'b', 'a']
```

5.3 列表的增加、删除、修改

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347145>

全套视频链接：<http://edu.51cto.com/sd/e9508>

5.3.1 列表元素的修改

```
1-> l=['a-黄瓜','b-南瓜','c-东瓜','d-苦瓜','e-西瓜']
2-> l[2]='c-我不是东瓜';print(l)
```

5.3.2 列表元素的增加

```
1-> l=['张三']
2-> l=['李四']+l;print(l)      #加号添加
3-> l.append('王二');print(l)  #append 添加
4-> l.insert(2,'麻子');print(l) #insert 插入添加
```

5.3.3 列表元素的删除

```
1-> #删除 1(按指定下标删除，没有该元素时报错)
2-> l2=['a-黄瓜','b-南瓜','c-东瓜','d-苦瓜','e-西瓜']
3-> l2.pop();print(l2)
4-> l2.pop(0);print(l2)

5-> #删除 2(按指定值删除)
6-> l3=['a-黄瓜','b-南瓜','c-东瓜','d-苦瓜','e-西瓜']
7-> l3.remove('c-东瓜');print(l3)

8-> #删除 3(del 语句删除)
9-> l4=['a-黄瓜','b-南瓜','c-东瓜','d-苦瓜','e-西瓜']
10-> del l4[-2];print(l4)

11-> #清空列表
12-> l5=['a-黄瓜','b-南瓜','c-东瓜','d-苦瓜','e-西瓜']
13-> l5.clear();print(l)
```

5.4 实例应用（汇总每个人的总成绩）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347144>

全套视频链接：<http://edu.51cto.com/sd/e9508>

#方法1

```
1-> import xlrd    #导入读取模块
2-> from xlutils.copy import copy #导入复制模块
3-> wb=xlrd.open_workbook('成绩统计表.xls') #读取工作簿
4-> ws=wb.sheet_by_name('成绩')    #读取工作表
5-> nwb=copy(wb);nws=nwb.get_sheet('Sheet2')    #复制工作簿，且读取工作表
6-> n=0;m=0 #初始化变量n,m
7-> while n<ws.nrows-1:    #循环行号
8->     n+=1    #累加数字
9->     l=ws.cell_value(n,1).split('-')[1::2] #拆分提取分数
10->     for v in l: #循环分数列表
11->         m+=int(v)#转换累加分数
12->     nws.write(n,0,ws.cell_value(n,0)) #将姓名写入A列单元格
13->     nws.write(n,1,m)    #将总分写入B列单元格
14->     m=0 #再次初始化变量m
15-> nwb.save('成绩统计表.xls') #保存工作簿
```

#方法2:

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> wb=xlrd.open_workbook('成绩统计表.xls')
4-> ws=wb.sheet_by_name('成绩')
5-> nwb=copy(wb);nws=nwb.get_sheet('Sheet2')
6-> col=ws.col_values(1)[1:]
7-> subtotal,n=0,0
8-> for v in col:
    a) for score in v.split('-')[1::2]:
        i. subtotal+=int(score)
    b) n+=1
    c) nws.write(n,0,ws.cell_value(n,0))
    d) nws.write(n,1,subtotal)
    e) subtotal=0
9-> nwb.save('成绩统计表.xls')
```

5.5 列表操作符

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347143>

全套视频链接：<http://edu.51cto.com/sd/e9508>

5.5.1 操作符基础

列表操作符有组合 (+)、重复 (*)、元素是否存在于列表中 (in), 下面就一一做介绍。

```
32-> print([1,2,3]+[100,200]) #组合
33-> print([1,2,3]*4)         #重复
34-> print(1 in [1,2,3])      #指定值在列表中是否存在
```

5.5.2 实例应用：判断指定姓名在指定日期是否值班

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> wb=xlrd.open_workbook('值班表.xls')
4-> ws=wb.sheet_by_name('值班表')
5-> nwb=copy(wb);nws=nwb.get_sheet('值班表')
6-> r=0
7-> while r<ws.nrows-1:
8->     r+=1
9->     row=ws.row_values(r) #读取要判断的行列表
10->    if '岳成周' in row: #判断岳成周在行列表中是否存在
11->        nws.write(r,6,'√') #如果存在则返回勾
12->    else:
13->        nws.write(r,6,'x') #如果不存在则返回叉
14-> nwb.save('值班表.xls')
```

5.6 列表推导式

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347142>

全套视频链接：<http://edu.51cto.com/sd/e9508>

列表推导式是 Python 基础，好用，而又非常重要的功能，也是最受欢迎的 Python 特性之一，可以说掌握它是成为合格 Python 程序员的基本标准。本质上可以把列表推导式理解成一种集合了转换和筛选功能的函数，通过这个函数把一个列表转换成另一个列表。注意是另一个新列表，原列表保持不变。

1. 列表推导

[表达式 for 变量 in 列表]

2. 列表嵌套推导

[表达式 for 变量1 in 列表1 for 变量2 in 列表2]

3. 条件列表推导

[表达式 for 变量 1 in 列表 if 条件]

```
1-> l1=['83','98','91','100']
2-> print([int(x) for x in l1]) #列表推导

3-> l2=[[98,86,74,85],[96,99,75,92],[69,93,87,85]]
4-> print([v for l in l2 for v in l]) #列表嵌套推导

5-> l3=[85,68,98,74,95,82,93,88,74]
6-> print([x for x in l3 if x>=90]) #条件列表推导
```

5.7 实例应用（筛选各工作表中符合条件的值）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347141>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('2018 年业绩表.xls')
3-> nwb=xlwt.Workbook(encoding='uft-8')
4-> nws=nwb.add_sheet('筛选结果')
5-> r=0
6-> for ws in wb.sheets(): #循环遍历工作表
7->     col=ws.col_values(1)[1:]#读取 B 列的金额数据
8->     l=[str(int(amount)) for amount in col if amount >= 20000]#循环判断 B 列的金
    额如果大于等于 2 万，则返回对应值。
9->     r+=1
10->     nws.write(r,0,ws.name)
11->     nws.write(r,1,'、'.join(l))
12-> nwb.save('筛选结果.xls')
```

5.8 列表转换

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347140>

全套视频链接：<http://edu.51cto.com/sd/e9508>

list, Python 的 list 函数可以将其他数据类型转换为列表类型，并返回转换后的列表。当参数为空时，list 函数可以创建一个空列表。

reverse, Python 列表 reverse() 方法对列表中的元素进行反向排序。

copy, copy 方法是用于复制列表元素的。

zip,

```
1-> #list 方法
2-> print(list('abc'))
3-> #reverse 方法
4-> l1=['Google','QQ','Taobao','Baidu']
5-> l1.reverse()
6-> print(l1)
7-> #copy 方法
8-> l2=[1,2,3,4,5,6]
9-> l3=l2.copy()
10-> print(l3)
11-> #zip 方法
12-> l4=[[1,2,3],[4,5,6]]
13-> l5=list(zip(*l4))
14-> print(l5)

15-> l6=[1,2,3]
16-> l7=[4,5,6]
17-> l8=list(zip(l6,l7))
18-> print(l8)

19-> print([x,y for x,y in zip(l6,l7)])
```

5.9 实例应用（统计出大于等于 2 万的记录到新表）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347139>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('2018 年业绩表.xls')
3-> nwb=xlwt.Workbook(encoding='uft-8')
4-> nws=nwb.add_sheet('筛选结果')
5-> r=0;n=0
6-> for ws in wb.sheets():
7->     col0=ws.col_values(0)[1:]
8->     col1=ws.col_values(1)[1:]
9->     l=[x,y for x,y in zip(col0,col1) if y>=20000]#筛选大于等于2 万的记录
10->     for l1 in l:
11->         n+=1
12->         nws.write(n,0,ws.name)
13->         nws.write(n,1,l1[0]);nws.write(n,2,l1[1])
14-> nws.write(0,0,'月份');nws.write(0,1,'日期');nws.write(0,2,'金额')
15-> nwb.save('筛选结果 888.xls')
```

5.10 列表常见统计方式 1

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347138>

全套视频链接：<http://edu.51cto.com/sd/e9508>

len 计数

max 最大值

min 最小值

sum 求和

实例应用：

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> wb=xlrd.open_workbook('工资表.xls')
4-> ws=wb.sheet_by_name('工资表')
5-> nwb=copy(wb);nws=nwb.get_sheet('工资表')
6-> n=0
7-> for cell in ws.col_values(1)[1:]:
8->     l=[int(n) for n in cell.split(' ')]
9->     l1=['总工资: ','最高工资: ','最低工资: ','月份数: ','平均工资: ']
10->     l2=[sum(l),max(l),min(l),len(l),sum(l)/len(l)]
11->     l3=['元','元','元','个月','元']
12->     l4=[x+str(y)+z for x,y,z in zip(l1,l2,l3)]
13->     n+=1
14->     nws.write(n,2,'\n'.join(l4))
15-> nwb.save('工资表.xls')
```


5.11 列表常见统计方式 2

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347137>

全套视频链接：<http://edu.51cto.com/sd/e9508>

1. Index

Python 列表 `index()` 方法用于从列表中找出某个对象第一个匹配项的索引位置，如果这个对象不在列表中会报一个异常。

`index()` 方法语法：

`L.index(obj[,start=0[,stop=len(L)]])`

`obj` 查找的对象。

`start` 可选参数，开始索引，默认为 0。（可单独指定）

`stop` 可选参数，结束索引，默认为列表的长度。（不能单独指定）

2. count

Python 列表 `count()` 方法用于统计某个元素在列表中出现的次数。

`count()` 方法语法：

`L.count(obj)`

`obj` 列表中统计的对象。

```
1-> import xlrd
2-> wb=xlrd.open_workbook('获奖名单.xls')
3-> ws1=wb.sheet_by_name('Sheet1')
4-> ws2=wb.sheet_by_name('统计表')
5-> n=0;l=[]
6-> while n<ws1.ncols-1:
7->     n+=1
8->     l+=ws1.col_values(n)[1:]
9-> l2=ws2.col_values(0)[1:]
10-> l3=[x+'!'+str(l.count(x))+'次' for x in l2]
11-> print(l3)
```

第6章 元组处理技术

元组被称为只读列表，数据可被查询，但不能被修改，元组写在小括号里面（）元素之间用逗号隔开。对于一些不想被修改的数据，可以用元组来保护。

6.1 元组的创建

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347136>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> #创建多个值的元组
2-> t1=(3,'a',[1,2,3],999)
3-> print(t1)

4-> #创建单个值的元组
5-> t2=(100,)
6-> print(t2)

7-> #创建空元组
8-> t3=()
9-> print(t3)

10-> #元组的转换
11-> t4=tuple([1,2,3,4]) #列表转元组
12-> print(t4)
13-> print(list(t4)) #元组转列表
```

6.2 元组的基本操作

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347135>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> #元组的连接
2-> t1=(1,2,3)
3-> t2=(100,200,300,400)
4-> t3=t1+t2
5-> print(t3)

6-> #元组的删除
7-> t4=(1,2,3)
8-> del t4
9-> # print(t4)

10-> #元组的切片（与列表的访问方式一样）
11-> t5=(1,2,3,99,100,235)
12-> print(t5[3])
```

6.3 元组常用统计方法

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347134>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> #常见元组的统计
2-> t=(50,60,74,63,50,95,74,80,50)
3-> print(len(t)) #计数
4-> print(max(t)) #最大值
5-> print(min(t)) #最小值
6-> print(sum(t)) #求和
7-> print(t.count(50)) #条件计数
8-> print(t.index(80)) #条件定位

9-> #元组的推导式
10-> t1=('销售部-张三','财务部-李诗','销售部-周立新','IT 部-曾贤志','销售部-梁芬')
11-> t2=(t for t in t1 if t[:3]=='销售部')
12-> print(tuple(t2))
```

第7章 字典处理技术

Python 字典是另一种可变容器模型，且可存储任意类型对象，如字符串、数字、元组等其他容器模型。

7.1 字典的基础

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347133>

全套视频链接：<http://edu.51cto.com/sd/e9508>

字典由键和对应值成对组成。字典也被称作关联数组或哈希表。基本语法如下：

```
d = {key1 : value1, key2 : value2 }
```

字典的每个键值 key=>value 对用冒号 (:) 分割，每个键值对之间用逗号 (,) 分割，整个字典包括在花括号 ({}) 中。

如果要创建空字典，有如下两种方法：

方法 1：空字典创建，使用 {} 或者 dict()

方法 2：使用 dict

7.1.1 创建字典

```
1-> d={'bob':98,'lucass':85,'lily':92}
2-> print(d)

1-> d1={}      #空字典创建方法1
2-> d1['a']=100 #在现有字典添加键/值对
3-> print(d1)
4-> d1['b']=200
5-> print(d1)
```

7.1.2 字典键的特性

键必须具有唯一性：

不允许同一个键出现两次。创建时如果同一个键被赋值两次，后一个值会被记住，如下实例：

```
1-> d1={'name': '曾贤志', 'age': 36, 'name': '小明'}
2-> print(d1)
```

键必须不可变：

键必须不可变，所以可以用数字、字符串、元组，所以用列表就不行，如下实例：

```
1-> d2={99:'数字','name':'字符串',(1,2):'元组',[99]:'不能为列表'}
2-> print(d2)
```

7.1.3 获取字典里的值

```
1-> d1={'bob':98,'lucass':85,'lily':92}
2-> print(d1['lucass'])
3-> print(d1['lili']) #如果字典键不存在则返回错误
```

7.1.4 小实例

```
35-> l=['张三','李四','张三','王二','张三','王二','李四','张三']
36-> d=dict()
37-> for x in l:
38->     d[x]="
39-> print(list(d))
```

7.2 实例应用（提取产品最后的记录）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347132>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd
2-> wb=xlrd.open_workbook('销售表.xls')
3-> ws=wb.sheet_by_name('Sheet1')
4-> r=0;d=dict()
5-> while r<ws.nrows-1:
6->     key=ws.cell_value(r,0)
7->     val=ws.cell_value(r,1)
8->     d[key]=val
9->     r+=1
10-> print(d)
```

7.3 字典的转换

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347131>

全套视频链接：<http://edu.51cto.com/sd/e9508>

7.3.1 dict 转换法

```
1-> d1=dict(a=1,b=2)      #创建方法 1
2-> print(d1)

3-> d2=dict([('a',1),('b',2)]) #创建方法 2
4-> print(d2)

5-> d3=dict([['a',1],['b',2]]) #创建方法 3
6-> print(d3)

7-> d4=dict([['a',1],('b',2)]) #创建方法 4
8-> print(d4)

9-> d5=dict([['a',1],['b',2]]) #创建方法 5
10-> print(d5)

11-> d6=dict(zip(['a','b','c'],[1,2,3])) #创建方法 6
12-> print(d6)
```

7.3.2 dict.fromkeys 转换法

第一个参数是一个列表或者元组，里边的值为 key，第二个参数是所有 key 的 value 值。

```
1-> d7=dict.fromkeys(['a','b'],1)
2-> print(d7)
3-> d8=dict.fromkeys(['a','b'],1)
4-> print(d8)
```

7.4 实例应用（多列求唯一值）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347130>

全套视频链接：<http://edu.51cto.com/sd/e9508>

统计省份、客户名称、订单编号三列中的唯一值。

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('业绩表.xls')
3-> ws=wb.sheet_by_name('Sheet1')
4-> nwb=xlwt.Workbook(encoding='uft-8')
5-> nws=nwb.add_sheet('sheet1')
6-> l=[ws.col_values(0),ws.col_values(1),ws.col_values(3)]
7-> d=dict.fromkeys(zip(*l))
8-> r=0
9-> for x in d:
10->     nws.write(r,0,x[0])
11->     nws.write(r,1,x[1])
12->     nws.write(r,2,x[2])
13->     r+=1
14-> nwb.save('结果表.xls')
```

7.5 字典的删除

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347129>

全套视频链接：<http://edu.51cto.com/sd/e9508>

7.5.1 Clear 方法清空字典

#clear() 方法（删除字典内所有元素）

```
1-> d1={'name':'曾贤志','age':28,'salary':9800}
2-> d1.clear()
3-> print(d1)
```

7.5.2 pop 方法删除字典

#pop() 方法（删除字典给定键 `key` 所对应的值，返回值为被删除的值）

```
1-> d2={'name': '曾贤志', 'age': 28, 'salary': 9800}
```

```
2-> d2.pop('name')
```

```
3-> print(d2)
```

#popitem() 方法（随机返回并删除字典中的一对键和值）

```
1-> d3={'name': '曾贤志', 'age': 28, 'salary': 9800}
```

```
2-> d3.popitem()
```

```
3-> print(d3)
```

7.5.3 del 语句删除字典

#del 全局方法（能删单一的元素也能清空字典，清空只需一项操作）

```
1-> d4={'name': '曾贤志', 'age': 28, 'salary': 9800}
```

```
2-> del d4['age']
```

```
3-> print(d4)
```

```
4-> del d4    #全部清除
```

```
5-> print(d4)
```

7.6 实例应用（查询未发货订单）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347128>

全套视频链接：<http://edu.51cto.com/sd/e9508>

订单工作簿中有三个表，“全部订单”工作表显示了所有订单，“已发货”显示已发货的订单，现在要求在“未分货”工作表中统计出未分货的订单。

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> wb=xlrd.open_workbook('订单.xls')
4-> ws1=wb.sheet_by_name('全部订单')
5-> ws2=wb.sheet_by_name('已发货')
6-> nwb=copy(wb)
7-> nws=nwb.get_sheet('未发货')
8-> l1=[ws1.col_values(0),ws1.col_values(1),ws1.col_values(2)]
9-> l2=[ws2.col_values(0),ws2.col_values(1),ws2.col_values(2)]
10-> d1=dict.fromkeys(zip(*l1))
11-> d2=list(dict.fromkeys(zip(*l2)))[1:]
12-> for x in d2:
13->     d1.pop(x)
14-> r=0
15-> for x in d1:
16->     nws.write(r,0,x[0])
17->     nws.write(r,1,x[1])
18->     nws.write(r,2,x[2])
19->     r+=1
20-> nwb.save('订单.xls')
```

7.7 字典的修改

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347127>

全套视频链接：<http://edu.51cto.com/sd/e9508>

7.7.1 字典的值修改

```
1-> d1={'bob':98,'lucass':85,'lily':92}
2-> d1['lily']=100
3-> print(d1)
```

7.7.2 字典的键名修改

```
1-> d2={'bob':98,'lucass':85,'lily':92}
2-> d2['lily']=d2.pop('lily')
3-> print(d2)
```

7.7.3 字典的连接

```
1-> d1={'name':'曾贤志'}
2-> d2={'age':54}
3-> d1.update(d2)
4-> print(d1)
```

7.8 实例应用（统计各种蔬菜的总金额）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347126>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
40-> import xlrd
41-> wb=xlrd.open_workbook('蔬菜统计表.xls')
42-> ws=wb.sheet_by_name('Sheet1')
43-> r=0;d=dict()
44-> while r<ws.nrows-1:
45->     r+=1
46->     item=ws.row_values(r)[1:]
47->     if item[0] in d.keys():    #判断键在字典中是否存在
48->         d[item[0]]+=item[1]    #如果存在则累加
49->     else:
50->         d[item[0]]=item[1]    #如果不存在则添加
51-> print(d)
```

7.9 字典的查询

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347125>

全套视频链接：<http://edu.51cto.com/sd/e9508>

7.9.1 判断是否存在

使用 `in dict.keys()` 判断指定的键名是否在字典中存在。

7.9.2 单值查询

```
52-> d={'name':'曾贤志','age':54,'sector':'财务部'}
53-> print(d['name'])#获取指定键返回的值 方法1
54-> print(d.get('name')) #获取指定键返回的值 方法2
55-> print(d.get('names','无名氏')) #不存在键对应返回默认值
```

7.9.3 多值查询

```
1-> d={'name':'曾贤志','age':54,'sector':'财务部'}
#keys 方法返回一个可迭代对象，用list 转换为列表，用tuple1 转为元组。
2-> print(list(d.keys()))
#values 方法返回一个可迭代对象，用list 转换为列表，用tuple1 转为元组。
3-> print(list(d.values()))
#items 方法返回一个可迭代对象，用list 转换为列表，用tuple1 转为元组。
4-> print(list(d.items()))
```

7.10 实例应用（统计各日期的销售数据）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347124>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd
2-> wb=xlrd.open_workbook('销售表.xls')
3-> ws=wb.sheet_by_name('Sheet1')
4-> r=0;d=dict()
5-> while r<ws.nrows-1:
6->     r+=1
7->     key=ws.cell_value(r,0)
8->     value=ws.cell_value(r,1)
9->     if key in d.keys():
10->         d[key]+=[int(value)]
11->     else:
12->         d[key]=[int(value)]
13-> for k in d:
14->     print(k,[sum(d[k]),max(d[k]),min(d[k]),len(d[k])])
```

7.11 字典的循环

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347123>

全套视频链接：<http://edu.51cto.com/sd/e9508>

7.11.1 普通循环

```
1-> d={'name':'曾贤志','age':54,'sector':'财务部'}
2-> for x in d:
3->     print(x) #默认循环读取键名

4-> for x in d.keys():
5->     print(x) #循环读取键名

6-> for x in d.values():
7->     print(x) #循环读取键对应的值

8-> for x in d.items():
9->     print(x) #循环读取键/值对

10-> for x,y in d.items():
11->     print(x,y) #分别循环读取键名和值
```

7.11.2 字典在列表中的推导

```
1-> d={'name':'曾贤志','age':54,'sector':'财务部'}
2-> print([x for x in d])
3-> print([x for x in d.keys()])
4-> print([x for x in d.values()])
5-> print([x,y for x,y in d.items()])
```

7.11.3 字典推导式

1.对已经存在的字典重新推导。

```
1-> d={'a':1,'b':3,'c':10}
2-> d1={y:x for x,y in d.items()}
3-> print(d1)
4-> d2={x:y+10 for x,y in d.items()}
5-> print(d2)
```

2.将其它结构的数据推导为字典

```
1-> l1=['张三','李四','周立']
2-> l2=[100,90,85]
3-> d={x:y for x,y in zip(l1,l2)}
4-> print(d)
```

7.12 实例应用（统计各省各公司总业绩）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347122>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('业绩表.xls')
3-> ws=wb.sheet_by_name('Sheet1')
4-> nwb=xlwt.Workbook(encoding='uft-8')
5-> r=0;d={}
6-> while r<ws.nrows-1:
7->     r+=1
8->     l=ws.row_values(r)
9->     if (l[0],l[1]) in d.keys():
10->         d[(l[0],l[1])]+=l[5]
11->     else:
12->         d[(l[0],l[1])]=l[5]
13-> l1=dict.fromkeys([x for x,y in d.keys()]).keys()
14-> r=0
15-> for s in l1:
16->     nws=nwb.add_sheet(s)
17->     nws.write(0,0,'省份');nws.write(0,1,'公司名');nws.write(0,2,'总金额')
18->     l2=[[x[0],x[1],y] for x,y in d.items() if x[0]==s]
19->     for val in l2:
20->         r+=1
21->         nws.write(r,0,val[0]);nws.write(r,1,val[1]);nws.write(r,2,val[2])
22->     r=0
23-> nwb.save('汇总表.xls')
```

第8章 集合处理技术

python 中的集合对象是一组无序排列的可哈希的值。集合成员可以做字典中的键。集合支持用 `in` 和 `not in` 操作符检查成员，集合本身是无序的，不可以为集合创建索引或执行切片操作，也没有键可用来获取集合中的值。

集合具有重去功能。

8.1 集合的创建

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347121>

全套视频链接：<http://edu.51cto.com/sd/e9508>

8.1.1 可变集合(set)

```
1-> s1={1,2,3,4} #创建集合1
2-> print(s1)
3-> s2=set([1,2,3]) #创建集合方法2
4-> print(s2)
5-> s3=set() #创建空集合
6-> print(s3)
```

注意：想要创建空集合，你必须使用 `set()` 而不是 `{}`。后者用于创建空字典，我们在后面介绍的一种数据结构。

8.1.2 不可变集合(frozenset)

```
1-> s1={1,2,3} #创建可变集合
2-> s2=frozenset({1,2,3}) #创建不可变集合
3-> print(type(s1),type(s2))
4-> print(s1,s2)

5-> d1={s1:100} #不能做字段的键
6-> d2={s2:100} #可做字典的键
```

对于可变集合，可以添加和删除元素；对于不可变集合只能读取元素，不能进行修改，可变集合不是可哈希的，因此不能用做字典的键也不能做其他集合中的元素。

8.2 实例应用（判断指定项目是否存在）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347120>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd,xlwt
2-> from xlutils.copy import copy
3-> wb=xlrd.open_workbook('兴趣班报名表.xls')
4-> ws=wb.sheet_by_name('Sheet1')
5-> nwb=copy(wb);nws=nwb.get_sheet('Sheet1')
6-> r=0
7-> while r<ws.nrows-1:
8->     r+=1
9->     v=ws.cell_value(r,1)
10->     if set(v.split(','))== {'钢琴','绘画','书法','武术'}:
11->         nws.write(r,2,'全参')
12->     else:
13->         nws.write(r,2,'未全参')
14-> nwb.save('sss.xls')
```

8.3 集合的添加与删除

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347119>

全套视频链接：<http://edu.51cto.com/sd/e9508>

8.3.1 集合的添加

```
1-> s1={1,2,3}
2-> s1.add('a')#向集合中添加元素
3-> print(s1)
4-> s1.update('b')#向集合中添加元素
5-> print(s1)
6-> s1.update(['c','d','e'])#向集合中添加多个元素
7-> print(s1)
```

8.3.1 集合的删除

```
1-> s={'曾贤志','郭芙晨','章雪优','刘德花','李明'}
2-> s.remove('章雪优')#如果不存在则返回错误
3-> print(s)
4-> s.discard('刘德花')#如果不存在则什么都不做
5-> print(s)
6-> s.pop()#删除集合中一个不确定的元素，如果为空引发KeyError 错误。
7-> print(s)
8-> s.clear()#清空集合中所有元素
9-> print(s)
```

8.4 实例应用（多行多列求唯一值）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347118>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('排名表.xls')
3-> ws=wb.sheet_by_name('名次表')
4-> nwb=xlwt.Workbook(encoding='uft-8');nws=nwb.add_sheet('统计结果')
5-> c=0;s=set()
6-> while c<ws.ncols-1:
7->     c+=1
8->     s1=set(ws.col_values(c)[1:])
9->     s.update(s1)
10-> l=list(s)
11-> for n in range(len(s)):
12->     nws.write(n+1,0,n+1)
13->     nws.write(n+1,1,l[n])
14-> nws.write(0,0,'编号');nws.write(0,1,'姓名')
15-> nwb.save('去重结果表.xls')
```

8.5 集合的循环与推导

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347117>

全套视频链接：<http://edu.51cto.com/sd/e9508>

8.5.1 集合的循环

集合的循环与列表、元组、字典的循环基本相似。

8.5.2 集合的推导

集合推导式跟列表推导式差不多，都是对一个列表的元素全部执行相同的操作，但集合是一种无重复无序的序列

集合的特性是：1.使用大括号；2.结果中无重复；3.结果是一个 `set()` 集合，集合里面是一个序列。

```
1-> l=[1,2,3,45]
2-> s={x+100 for x in l}
3-> print(s)
4-> print(set(l))
5->
6-> s={2,3,54,65}
7-> s1={x*100 for x in s}
8-> print(s1)
```

8.6 实例应用（统计每月每个战队的人数）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347116>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('各战队比赛名次.xls')
3-> ws=wb.sheet_by_name('2018 年各战队排名')
4-> nwb=xlwt.Workbook(encoding='uft-8');nws=nwb.add_sheet('结果')
5-> r=0
6-> while r<ws.nrows-1:
7->     r+=1
8->     l=ws.row_values(r)
9->     l1=[x.split('-')[1] for x in l[1].split(',')]
10->     l2=[l[0]+'/'.join({'%s:%d 人'%(x,l1.count(x)) for x in l1})]
11->     nws.write(r,0,l2[0]);nws.write(r,1,l2[1])
12-> nwb.save('统计结果.xls')
```

8.7 集合的运算

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347115>

全套视频链接：<http://edu.51cto.com/sd/e9508>

python 的集合和其他语言类似，是一个无序不重复元素集。集合对象还支持 union(并), intersection(交), difference(差)和 symmetric.difference(对称差集)等数学运算。

```
1-> s1={'张三','王二麻子','李四','大周','小明'}
2-> s2={'李四','黄忠','刘精','王二麻子'}
3->
4-> #交集
5-> print(s1.intersection(s2))
6-> print(s1&s2)
7->
8-> #并集
9-> print(s1.union(s2))
10-> print(s1|s2)
11->
12-> #差集
13-> print(s1.difference(s2))
14-> print(s1-s2)
15-> print(s2.difference(s1))
16-> print(s2-s1)
17->
18-> #对称差集
19-> print(s1.symmetric_difference(s2))
20-> print(s2.symmetric_difference(s1))
21->
22-> #集合判断
23-> s3={'张三','李四'}
24-> s4={'张三','李四'}
25-> print(s3.issubset(s1))#判断s3 中的集合是否在s1 中全部存在
26-> print(s1.issuperset(s3))#判断s1 中是否包含s3 的全部内容
27-> print(s3.isdisjoint(s4))#如两个集合中有相同元素返回False， 否则返回True
```

8.8 实例应用（求每个人不达标的月份）

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347114>

全套视频链接：<http://edu.51cto.com/sd/e9508>

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('业绩达标表.xls');ws=wb.sheet_by_name('业绩')
3-> nwb=xlwt.Workbook(encoding='uft-8');nws=nwb.add_sheet('结果')
4-> col1=ws.col_values(0)[1:]
5-> col2=ws.col_values(1)[1:]
6-> s=set(zip(col1,col2))
7-> s1={'%d 月'%m for m in range(1,13)}
8-> s2=set(col1)
9-> r=0
10-> for m in s2:
11->     s3='/''.join(s1-{y for x,y in s if m==x})
12->     nws.write(r,0,m);nws.write(r,1,s3);r+=1
13-> nwb.save('统计结果.xls')
```


第9章 Python 函数技术

函数是 python 编程中的核心内容之一。函数的最大优点是：增强代码的重用性和可读性。Python 中有很多内置函数，但并不能满足用户需求，所以本章将学习如何自定义函数。

9.1 自定义函数的基本结构

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347113>

全套视频链接：<http://edu.51cto.com/sd/e9508>

9.1.1 函数的定义

在 Python 中，函数有五大要点，分别是 def、函数名、函数体、参数、返回值，分别是括号（括号内为参数）和冒号（:）。

def：函数的关键字，没它可不行。

函数名：函数的名称，根据函数名调用函数。

函数体：函数中进行一系列的具体操作。

参数：为函数体提供数据。

返回值：当函数执行完毕后，可以给调用者返回数据。Return 编写要返回的值，也是函数结束的标志，return 的返回值个数无限制，即可以用逗号分隔开多个任意类型的值。

函数定义书写格式如下：

```
def 函数名(参数):  
    函数体  
    return 返回值
```

无参函数

```
1-> import os  
2-> def path():  
3->     return os.getcwd()  
4-> print(path())#应用函数
```

有参函数

```
1-> def avg(list):  
2->     val=sum(list)/len(list)  
3->     return val  
4-> print(avg([4,5])) #应用函数
```

9.1.2 实例应用（定义平均函数及应用）

要求：提取身份证中的日期

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> #-----
4-> def age(id,dit):
5->     l=[id[x:y] for x,y in ((6,10),(10,12),(12,14))]
6->     date=dit.join(l)
7->     return date
8-> #-----
9-> wb=xlrd.open_workbook('用户信息.xls')
10-> ws=wb.sheet_by_name('信息表')
11-> nwb=copy(wb);nws=nwb.get_sheet('信息表')
12-> r=0
13-> while r<ws.nrows-1:
14->     r+=1
15->     nws.write(r,2,age(ws.cell_value(r,1),'-'))
16-> nwb.save('用户信息.xls')
```

9.2 位置参数写法及应用

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347112>

全套视频链接：<http://edu.51cto.com/sd/e9508>

9.2.1 位置参数（必需参数）

位置参数，即函数调用时必须传入的参数。一个也不能缺。

```
1-> #-----定义字符串提取函数-----
2-> def mid(text,start,num):
3->     txt=text[start:start+num]
4->     return txt
5-> #-----
6-> s='QQ:546502445'
7-> print(mid(s,3,5))#使用定义的函数提取字符串
```

9.2.2 实例应用（等级判断函数及应用）

要求：根据定义好的等级函数判断提供分数的等级。

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> #-----
4-> def level(num,l1,l2,l3,l4):
5->     if num>=l1:
6->         l='优'
7->     elif num>=l2:
8->         l='良'
9->     elif num>=l3:
10->         l='中'
11->     elif num>=l4:
12->         l='差'
13->     return l
14-> #-----
15-> wb=xlrd.open_workbook('分数表.xls')
16-> ws=wb.sheet_by_name('信息表')
17-> nwb=copy(wb);nws=nwb.get_sheet('信息表')
18-> r=0
19-> for c in ws.col_values(1)[1:]:
20->     r+=1
21->     nws.write(r,2,level(c,90,85,70,0))
22-> nwb.save('分数表.xls')
```

9.3 默认参数写法及应用

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347111>

全套视频链接：<http://edu.51cto.com/sd/e9508>

9.3.1 默认参数（可选参数）

用于定义函数，为参数提供默认值，调用函数时可传可不传该默认参数的值（注意：所有位置参数必须出现在默认参数前，包括函数定义和调用）

```
1-> l=[['a',1,1],['b',3,1],['c',5,10],['a',1,10],['c',1,10]]
2-> def distinct(l,num=0):
3->     return {x[num] for x in l}
4-> print(distinct(l))
```

9.3.2 实例应用（定位指定字符串的位置）

要求：提取指定分隔符后的字符串。

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> #-----
4-> def Text_pos(str,delimiter,position=0):
5->     n,l=1,[-1]
6->     while n<=str.count(delimiter):
7->         l.append(str.index(delimiter,l[-1]+1))
8->         n +=1
9->     l = l[1:]
10->     if position==0:
11->         l=l[0]
12->     elif position==1:
13->         l=l[-1]
14->     elif position==2:
15->         l=l[:]
16->     return l
17-> def mid(text,start,num):
18->     txt=text[start:start+num]
19->     return txt
20-> #-----
21-> wb=xlrd.open_workbook('产品信息表.xls')
22-> ws=wb.sheet_by_name('产品表')
23-> nwb=copy(wb)
24-> nws=nwb.get_sheet('产品表')
25-> n=0
26-> for c in ws.col_values(1)[1:]:
27->     n +=1
28->     l=[mid(' '+c,x+1,6) for x in Text_pos(' ', 'c', '2)]
29->     nws.write(n,2,' '.join(l))
30-> nwb.save('产品信息表.xls')
```

9.4 关键字参数写法及应用

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347110>

全套视频链接：<http://edu.51cto.com/sd/e9508>

用于函数调用，通过“键-值”形式加以指定。可以让函数更加清晰、容易使用，同时参数的编写可以不按顺序。

9.4.1 关键字参数

关键字参数只是一书写形式，并非参数形式。

```
1-> def mid(txt,start,le=1):
2->     s=txt[start:start+le]
3->     return s
4-> print(mid(txt='Python 与 Excel',le=5,start=2))
```

9.4.2 实例应用（汇总字符串中指定位置的数字）

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> #-----
4-> def agg(list,delimiter='-',subtotal=sum):
5->     l=[int(x.split(delimiter)[1]) for x in list]
6->     return subtotal(l)
7-> #-----
8-> wb=xlrd.open_workbook('2018 年业绩表.xls')
9-> ws=wb.sheet_by_name('2018 业绩表')
10-> nwb=copy(wb);nws=nwb.get_sheet('2018 业绩表')
11-> r=0
12-> while r<ws.nrows-1:
13->     r+=1
14->     l=ws.row_values(r,1)
15->     s=agg(l, subtotal=max, delimiter=',')
16->     nws.write(r,6,s)
17-> nwb.save('ssd.xls')
```

9.5 不定长参数写法及应用

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347109>

全套视频链接：<http://edu.51cto.com/sd/e9508>

9.5.1 不定长参数

不定长参数分为两种，一种为：`*args`，向这种参数写入的值是存储在元组中的。另一种为：`**args`，向这种参数写入的值是存储在字典中的。


```
1-> #不定长参数实例1
2-> def agg1(list,*subtal):
3->     l = [x(list) for x in subtal]
4->     return l
5-> print(agg1([3,34,4,3,5],sum,max,min))
6->
7-> #不定长参数实例2
8-> def agg2(list,**subtal):
9->     l=[(x,y(list)) for x,y in subtal.items()]
10->    return l
11-> print(agg2([3,34,4,3,5],求和=sum,求最大=max,求最小=min))
```

9.5.1 实例应用（增强替换函数的功能）

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> #-----
4-> def substitute(txt,*repl):
5->     for x,y in repl:
6->         txt=txt.replace(x,y)
7->     return txt
8-> #-----
9-> wb=xlrd.open_workbook('成绩表.xls')
10-> ws=wb.sheet_by_name('分数表')
11-> nwb=copy(wb)
12-> nws=nwb.get_sheet('分数表')
13-> n=0
14-> for c in ws.col_values(1)[1:]:
15->     n +=1
16->     nws.write(n,2,substitute(c,('(','(',')','(',')',' ','-'),(' ','\n')))
17-> nwb.save('成绩表.xls')
```

9.6 匿名函数写法及应用

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347108>

全套视频链接：<http://edu.51cto.com/sd/e9508>

9.6.1 匿名函数

没有名字的函数，不用写 `return`，返回值就是该表达式的结果。

1. 程序一次性使用，所以不需要定义函数名，节省内存中变量定义空间
2. 如果想让程序更加简洁时。

语法：`lambda` 参数：方法（或三元运算）

如 `f = lambda a,b,c:a+b+c` 中的关键字 `lambda` 表示匿名函数，冒号之前的 `a,b,c` 表示它们是这个函数的参数。匿名函数不需要 `return` 来返回值，表达式本身结果就是返回值。

```
1-> #无参
2-> d=lambda : 'test'
3-> print(d())
4-> #位置参数
5-> d1=lambda x,y,z:x+y+z
6-> print(d1(4,5,6))
7-> #默认参数
8-> d2=lambda x,y,z=100:x+y+z
9-> print(d2(2,3))
10-> #可变参数
11-> d3=lambda x,*y:x(y)
12-> print(d3(sum,1,2,3))
13-> #可变参数
14-> d4=lambda x,*y:[(n,m(x)) for n,m in y.items()]
15-> print(d4([2,34,3,5],求和=sum,最大值=max))
16-> #三元运算
17-> d5=lambda x,y:'成功' if x>=y else '失败'
18-> print(d5(18,60))
```

9.6.2 实例应用：根据身份证号判断性别

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> wb=xlrd.open_workbook('员工表.xls')
4-> ws=wb.sheet_by_name('信息')
5-> nwb=copy(wb)
6-> nws=nwb.get_sheet('信息')
7-> n=0
8-> for c in ws.col_values(1)[1:]:
9->     n +=1
10->     nws.write(n,2,(lambda id:'男' if int(id[-2])%2==1 else '女')(c))
11-> nwb.save('员工表.xls')
```

9.7 递归函数写法及应用

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347107>

全套视频链接：<http://edu.51cto.com/sd/e9508>

递归函数特性：

必须有一个明确的结束条件；

每次进入更深一层递归时，问题规模相比上次递归都应有所减少

相邻两次重复之间有紧密的联系，前一次要为后一次做准备（通常前一次的输出就作为后一次的输入）。

递归效率不高，递归层次过多会导致栈溢出

```
1-> #递归函数1
2-> def fact(x):
3->     if x==1:
4->         return 1
5->     else:
6->         return x+fact(x-1)
7->
8-> print(fact(5))
9->
10-> #递归函数2
11-> def con(l):
12->     if len(l)==0:
13->         return ""
14->     else:
15->         return con(l[:len(l)-1])+'-'+l[-1]
16->
17-> print(con(['a','b','c','d']))
```

第10章 高阶函数应用

高阶函数，就是把函数当成参数传递的一种函数。

10.1 map 转换函数

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347106>

全套视频链接：<http://edu.51cto.com/sd/e9508>

10.1.1 map 函数的应用方法

map()函数接收两个参数，一个是函数，一个是序列，map 将传入的函数依次作用到序列的每个元素，并把结果作为新的 list 返回。

```
1-> def fun(x):
2->     return x**2
3-> #方法 1
4-> l=[fun(x) for x in [2,4,7,5]]
5-> print(l)
6-> #方法 2
7-> print(list(map(fun,[2,4,7,5])))
8-> #方法 3
9-> print(list(map(lambda x:x**2,[2,4,7,5])))
10-> #方法 4
11-> print(list(map(len,['677','89','766756','9871'])))
```

10.1.2 实例应用（每个月大于等于 100 的记录数）

```
1-> import xlrd,xlwt
2-> wb=xlrd.open_workbook('业绩表.xls')
3-> nwb=xlwt.Workbook(encoding='utf-8')
4-> nws=nwb.add_sheet('结果')
5-> def counter(list):
6->     return len([x for x in list if x>=100])
7-> n=0
8-> for s in wb.sheets():
9->     n +=1
10->     nws.write(n,s.name)
11->     nws.write(n,1,list(map(counter,[s.col_values(1)[1:]]))[0])
12-> nws.write(0,0,'月份')
13-> nws.write(0,1,'计数')
14-> nwb.save('统计结果.xls')
```

10.2 filter 筛选函数

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347105>

全套视频链接：<http://edu.51cto.com/sd/e9508>

`filter()`函数用于过滤序列，过滤掉不符合条件的元素，返回由符合条件元素组成的新列表。

接收两个参数，第一个为函数，第二个为序列，序列的每个元素作为参数传递给函数进行判断，返回 True 或 False，将返回 True 的元素放到新列表中。

`filter(function, iterable)`

参数：

`function`：判断函数

`iterable`：可迭代对象

例子 1

```
1-> l=[45,89,67,56,98,81,44]
2-> #列表推导筛选
3-> l1=[x for x in l if x>80]
4-> print(l1)
5-> #自定义函数筛选
6-> def test1(x):
7->     return x>=80
8-> l2=filter(test1,l)
9-> print(list(l2))
10-> #匿名函数筛选
11-> l3=filter(lambda x:x>80,l)
12-> print(list(l3))
```

例子 2

```
13-> s=[[56,88,95],[81,93],[44,98],[99,87,100]]
14-> #列表推导筛选
15-> s1=[x for x in s if len(x)>2]#推导列表
16-> print(s1)
17-> #自定义函数筛选
18-> def test2(x):
19->     return len(x)>2
20-> s2=filter(test2,s)
21-> print(list(s2))
22-> #匿名函数筛选
23-> s3=filter(lambda x:len(x)>2,s)#filter 高阶函数
24-> print(list(s3))
```

例子 3

```
25-> n=[[89,69],[78,96],[85,99],[96,89]]
26-> #列表推导筛选
27-> n1=[[x,y] for x,y in n if x>=y]
28-> print(n1)
29-> #自定义函数筛选
30-> def test3(x):
31->     return x[0]>x[1]
32-> n2=filter(test3,n)
33-> print(list(n2))
34-> #匿名函数筛选
35-> n3=filter(lambda x:x[0]>=x[1],n)
36-> print(list(n3))
```


10.3 Sort 与 Sorted 排序函数

本节课程链接：<http://edu.51cto.com/center/course/lesson/index?id=347104>

全套视频链接：<http://edu.51cto.com/sd/e9508>

10.3.1 sort 方法

内建方法 `sort()`，可以直接对列表进行排序。语法如下：

`list.sort(key=None, reverse=False(or True))`

`key`，用户可以用函数来确定比较方式。

`Reverse`，这个 `bool` 类型参数，当 `reverse=False` 时：为正向排序；当 `reverse=True` 时：为反向排序。默认为 `False`。

执行完后会改变原来的 `list`

```
1-> #普通比较方式
2-> l1=[354,275,3,5,765,43]
3-> l1.sort(reverse=True)
4-> print(l1)
5-> l1.sort(reverse=False)
6-> print(l1)

7-> #自定义比较方式
8-> l2=['张三-98','李四-93','刘飞龙-87','曾贤志-100']
9-> l2.sort(key=lambda x:int(x.split('-')[1]),reverse=True)
10-> print(l2)
```

10.3.2 sorted 函数

`sorted()`不会改变原来的 `list`，而是会返回一个新的已经排序好的 `list`。`list.sort()`方法仅仅被 `list` 所定义，`sorted()`可用于任何一个可迭代对象。执行完后会有返回一个新排序好的 `list`

`Sorted` 函数语法：

`sorted(iterable, key, reverse)`

`iterable`，任何可迭代的对象

`key`，用户可以用函数来确定比较方式。

`reverse`，该函数也含有 `reverse` 这个 `bool` 类型的参数，当 `reverse=False` 时：为正向排序(从小到大)；当 `reverse=True` 时：为反向排序(从大到小)。当然默认为 `False`。

```
1-> #普通比较方式
2-> l1=[354,275,3,5,765,43]
3-> l2=sorted(l1,reverse=True);print(l2)
4-> l3=sorted(l1,reverse=False);print(l3)
5->
6-> #自定义比较方式
7-> l4=['张三-98','李四-93','刘飞龙-87','曾贤志-100']
8-> l5=sorted(l4,key=lambda x:int(x.split('-')[1]),reverse=True)
9-> print(l5)
10-> l6=sorted(l4,key=lambda x:int(x.split('-')[1]),reverse=False)
11-> print(l6)
```

10.3.3 实例应用（自定义排序）

```
1-> import xlrd
2-> from xlutils.copy import copy
3-> wb=xlrd.open_workbook('业绩表.xls')
4-> ws=wb.sheet_by_name('数据')
5-> nwb=copy(wb)
6-> nws=nwb.get_sheet('数据')
7-> l=list(zip(ws.col_values(0)[1:],ws.col_values(1)[1:]))
8-> ll=sorted(l,key=lambda x:int(x[1].split(':')[1]),reverse=True)
9-> n=0
10-> for c in ll:
11->     n+=1
12->     nws.write(n,0,c[0])
13->     nws.write(n,1,c[1])
14-> nwb.save('业绩表.xls')
```